1. INTRODUCTION

This document describes the administration functions performed in order to operate and maintain the Automated Message Handling System (AMHS), a subsystem of the Global Command and Control System (GCCS), to ensure continuous system availability. The administration functions are performed by AMHS System Administrators, and System Operators, in concert with other GCCS Administrators. The intent of this document is to provide a basic understanding of the system and descriptions of the functions and recommended processes that require operator/administrator attention. It can be used as a source for deriving lower level standard operating procedures (SOPs) that are consistent with the site-specific installation.

The AMHS segment is the Automatic Digital Network (AUTODIN) message processing component of the GCCS common operating environment (COE) core services consisting of process-unique software modules, and shares use of several commercial off-the-shelf (COTS) software components.

The GCCS is an Automated Information System (AIS) supporting the Department of Defense (DoD) Command and Control Mission. GCCS is producing, integrating, and fielding new hardware and software components designed to provide the Joint Planning and Execution Community (JPEC) with new technology and functionality. GCCS system integration emphasizes use of COTS products, and merges the capabilities of a modern Local Area Network (LAN), UNIX-based client/server architecture, desktop-style Graphical User Interface (GUI), and a Relational Database Management System (RDBMS).

GCCS is intended to help joint operation planners satisfy their deliberate and crisis planning responsibilities via access to a useful, user-tested, integrated set of analytic tools and flexible data transfer capabilities. The GCCS client/server architecture provides a firm foundation for linking external systems and GCCS components, permitting easy access to applications, and faster, more reliable data transfers within a secure environment. At the heart of GCCS is a large database and application server connected to a LAN. The GCCS LAN interconnects the GCCS server with a variety of workstations (Disk Operating System (DOS) and Microsoft Windows personal computers (PCs), Macintosh, UNIX, and other X Windows clients) that run associated software and application packages. The GCCS LAN will also connect with Wide Area Networks (WANs) supporting standard LAN design.

The GCCS architecture is specifically designed with flexibility and COTS standardization to allow interconnection with new networks and systems as they are deployed. This architecture will easily adapt to and assimilate new applications and functions, ensuring cost effective migration to new technologies as they become available and potentially negating an otherwise inevitable obsolescence.

GCCS is designed with the user in mind and is a powerful and flexible, yet fully functional, set of tools. Achieving these goals involves a complex system design, with a regular and effective technical, “behind the scenes” system administration activity. Consequently, trained system administration personnel are *essential* to the satisfactory operation of the GCCS system resources at each site. This AMHS System Administration Manual provides technical system administration guidance for DoD sites receiving GCCS AMHS Version 2.1.

Figure 1-1 shows how the AMHS fits within the GCCS COE.

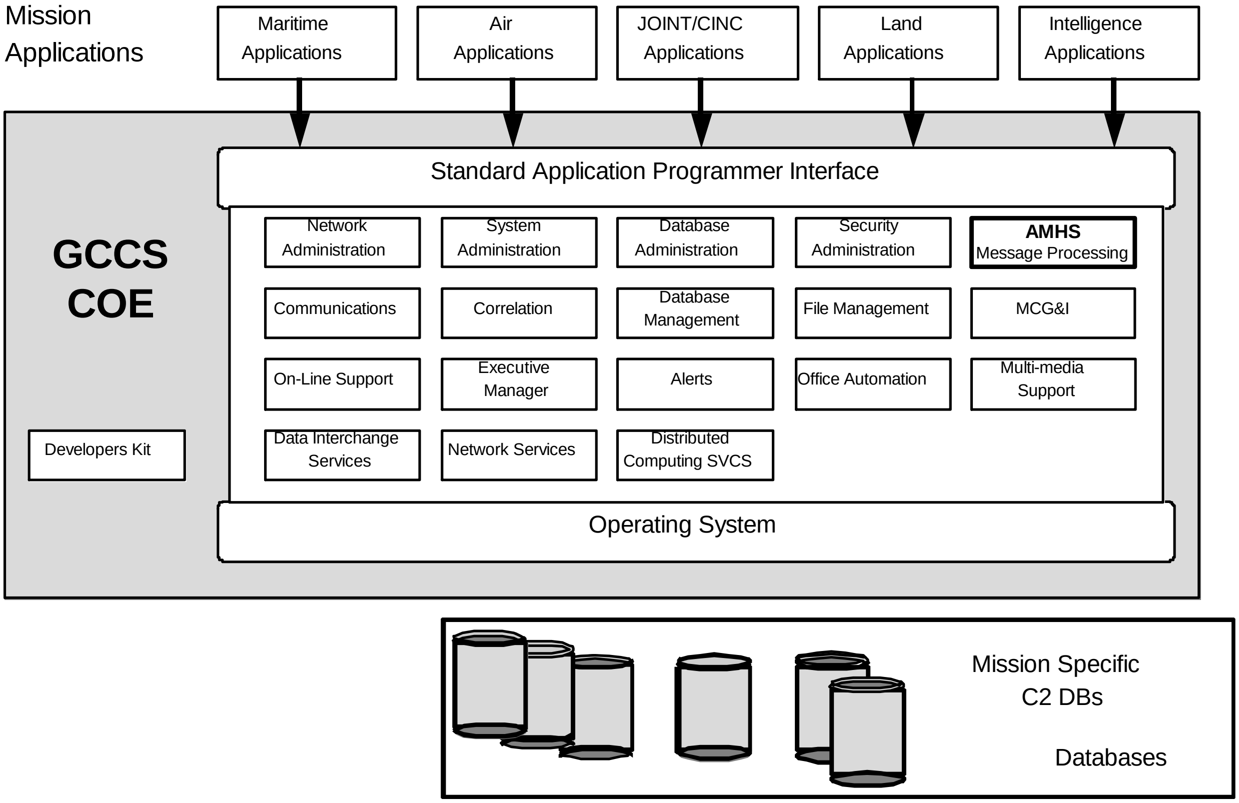


Figure 1-1. GCCS Common Operating Environment

1.1 AMHS OVERVIEW

The Automated Message Handling System (AMHS) of the Global Command and Control System (GCCS) provides three basic functions:

(1) Automated receipt, storage, and distribution directly to the User, of AUTODIN messages.

(1) Retrospective search and recall of stored messages (60 days, configurable). This is accomplished at the user workstation with simple standard Verity Topic query statements.

(2) Support for generating, verifying, approving and transmitting AUTODIN messages.

The AMHS is integrated into the GCCS COE utilizing several other common operating environment functions. It consists of several COTS software packages and programs that run on a heterogeneous network of Sun SPARC Servers, SPARC and Apollo workstations, and PC DOS-based hardware platforms as shown in Figure 1-2.

The AMHS COTS software packages are Verity TOPIC, Applix Words, and Sybase SQL Server for account authentication. The COEs are UNIX Solaris 2.3, Executive Manager, Command Center Applications, AMHS Server and AMHS Client.

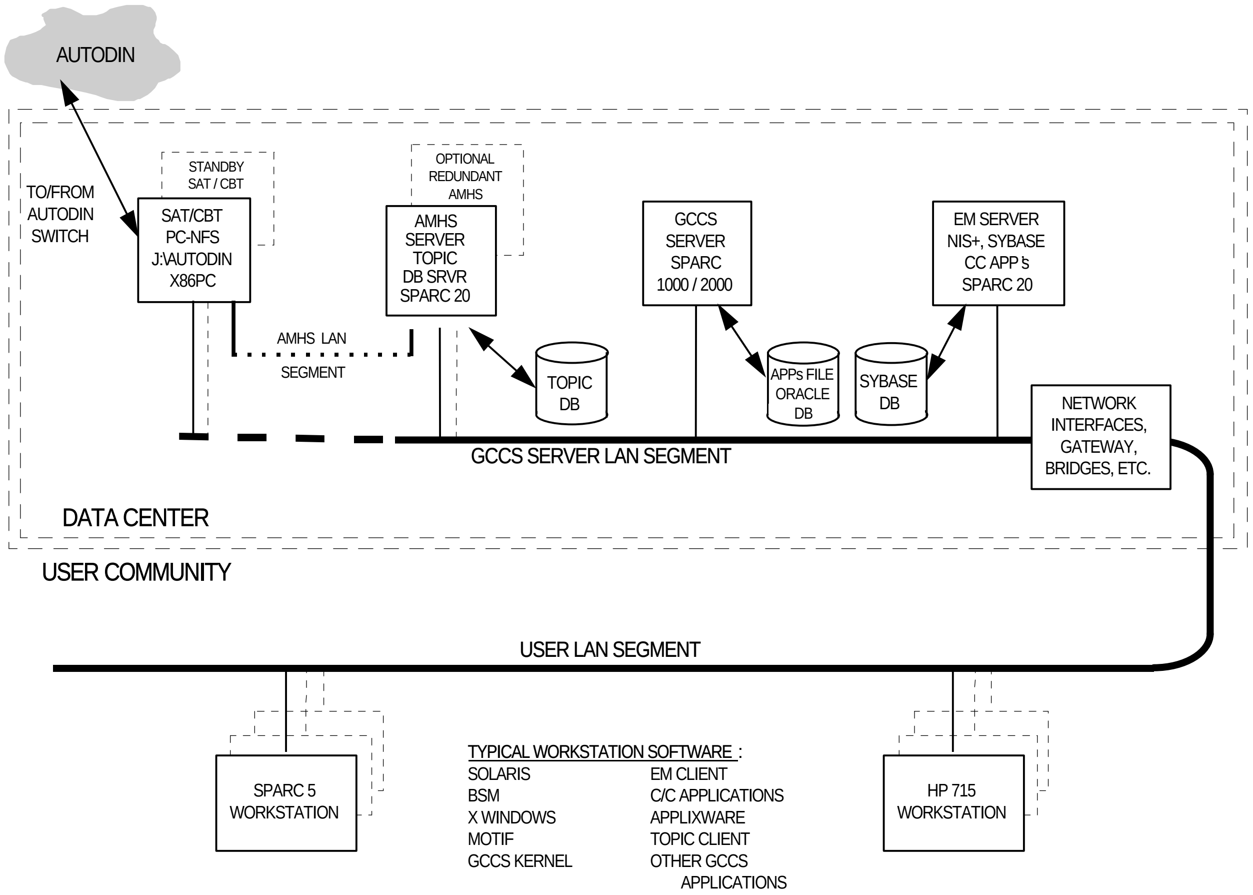


Figure 1-2. Typical GCCS Environment

1.2 PURPOSE

The purpose of this document is to provide necessary operational and technical reference information for GCCS personnel performing AMHS system activities under the managerial oversight of the GCCS Information Systems Security Officer (ISSO) and GCCS Information Systems Operations Officer (ISOO) or their appointed representatives in cooperation with the GCCS UNIX System Administrator, the LAN Administrator, the GCCS Systems Administrator and the GCCS Security Manager. These organizational positions are generic, and your organization may be structured consistent with Section 3.4.8.3 of the GCCS System and Network Management CONOPS V.1.6 document.

The inherent flexibility of the GCCS concept allows hardware and software to be configured to meet the specific needs of each site. This requires that support documentation be generic enough to cover all the possible configurations while meeting the needs of each site’s personnel. The job/role definitions used here are functional and do not necessarily mandate site staffing. It is essential that each site develop formal SOPs that translate the procedures, processes, methods, tasks and responsibilities outlined into site-specific processes and accountabilities to ensure system performance.

1.3 SCOPE

The GCCS AMHS is maintained by a group of people sharing the responsibility for system security, system operation, system administration and database administration. This document describes the activities performed by this group of people in order to ensure a continuously operational AMHS which is vital to the daily activities of the typical command center. The focus is on AMHS-specific concepts, processes and procedures. The interdependencies on the GCCS COE, and specifically EM Desktop tools and processes, will be discussed briefly as required, but familiarity and reference depends on these components’ documentation. Additional monitoring capabilities of the AMHS subsystem are available through the EM System Monitor and Control component program.

This document assumes a level of training and experience consistent with the assigned responsibility. Sections 4, 5 and 6 will delineate this for each area of responsibility. The Administrators will have one year UNIX Operating System (OS) experience, GCCS overview, and AMHS-specific training. The Operators will be familiar with UNIX commands and system operation principles.

1.4 DOCUMENT ORGANIZATION

The GCCS AMHS Administration Manual consists of the following seven sections and two appendices:

(1) Section 1, Introduction. Provides an overview of the AMHS subsystem and describes the manual, its purpose, use, organization, and list of applicable documents.

(2) Section 2, AMHS Overview and Theory of Operation. Provides functional descriptions of how the AMHS handles AUTODIN traffic to and from the user workstation.

(3) Section 3, AMHS Baseline Configuration. Provides the necessary information to plan, configure and operate a typical installation.

(4) Section 4, System Operator Tasks/Functions. Provides a description of the AMHS System Operator's activities and how those activities are performed.

(5) Section 5, System Administrator Reference. Provides a description of the AMHS System Administrator's activities and how those activities are performed.

(6) Section 6, Redundant AMHS Procedures. Contains the procedures for switching from primary to secondary string and other redundant operation subjects.

(7) Appendix A, Acronyms/Glossary. Contains a document-specific list of acronyms and a glossary.

(8) Appendix B, Installation and Configuration Notes. Contains reference material.

(9) Appendix C, GCCS System Admin Reference. Contains details on the use of Security Manager, Profile Manager, System Monitor, System Controller and System Administration.

1.5 APPLICABLE DOCUMENTS

This is not a stand-alone document as to the complete workings of the GCCS AMHS. Administrators should also be acquainted with the following publications as well as all site-specific documentation, e.g. your Site Description document.

1.5.1 Government Documents

(1) System User’s Manual: GCCS Ver. 2.1, CM #LL-500-133-01.[*]

(2) System Administration Manual: GCCS-SAM 2.1, CM #LL-500-29-06.[*]

(3) GCCS Automated Information Plan for Vers. 2.1, CM #LL-500-67-04.[*]

(4) GCCS Version Description Document, CM #LL-500-102-05.[*]

(5) GCCS Implementation Procedures for AIS, GCCS Version 2.1 rev 4, CM #LL-500-103-17.[*]

(6) GCCS Ad Hoc Query User Manual with Change Pages Inserted, CM #LL-500-147-03.[*]

(7) GCCS System Security Implementation Instructions for Site Security Administrators, LL-500-43-04.[*]

(8) GCCS System and Network Management CONOPS V.1.6, LL-500-189-01.[*]

(9) Automatic Digital Network (AUTODIN) Operating Procedures, JANAP 128 (I).

[*] Global Command and Control System Document Library, Defense Information Systems Agency.

[*] Global Command and Control System Document Library, Defense Information Systems Agency.

1.5.2 Non-Government Documents

(1) Security Description for the Jet Propulsion Laboratory Automated Message Handling System, JPL D-12076, Release 3.2, November 1994.

(2) Automated Message Handling System (AMHS) Defense Message System Functional Test Plan, JPL D-12615, Release 2, Change 1, 5 October 1995.

(3) Automated Message Handling System (AMHS) Security Test and Evaluation Plan for Solaris, JPL D-12575, Release 2, Change 1, 5 October 1995.

(4) Automated Message Handling System (AMHS) Security Analysis for Solaris, JPL D-12576, Release 1, 8 May 1995.

(5) Trusted Facility Manual Template, JPL D-12613, Release 1, 12 May 1995.

(6) GCCS Automated Message Handling System Application Programming Interface, JPL D-12731, Release 1, 5 October 1995.

(7) GCCS Executive Manager Application Programming Interface, JPL D-13141, Release 1, 20 January 1996.

(8) Standard Automated Terminal (SAT) System User's Manual, Cavalier Communications, Inc., September 1993.

(9) Communications Support Processor Backside Terminal (CBT) System User's Manual, Cavalier Communications, Inc., March 1994.

(10) Applixware Suite of Manuals; Applix Inc., Westboro, Mass.

(11) System and Network Administration, Volumes 1 and 2, Sun Part No. 800-3805-10, Sun Microsystems, Rev. A, 27 March 1990.

(12) Sybase SQL Server System Administrator Guide, Sybase Inc., 1 February 1994.

(13) SPARCPrinter II Hardware Installation and User's Guide, Sun Part No. 801-5806-10, Sun Microsystems, March 1994.

(14) TOPIC Database Administrator's Guide V3.1 Vol.1, Vol.2, Part No. NA-TOP-01-01, Verity Inc., 1992.* (Verity Inc., 1550 Plymouth Street, Mountain View, CA

(15) TOPIC REAL-TIME Administrator's Guide V3.1, Part No. NA-TOP-02-01, Verity Inc., 1992.*

(16) TOPIC Motif User's Guide V3.1, Verity Inc., 1992.

* Global Command and Control System Document Library, Defense Information Systems Agency.

2. AMHS OVERVIEW AND THEORY OF OPERATION

One of the key elements to planning and executing all military operations is communications. Information must flow quickly and reliably between all components of an organization, and in many cases between organizations. Historically the communications infrastructure is a cadre of skilled people committed to ensuring message delivery.

When an individual has information that they are responsible and accountable for, they must either act on the information or transfer the accountability to someone who can act. The Telecommunications Centers (TCC) AUTODIN infrastructure and skilled operators facilitate the transfer of information and accountability, or return the message to the originator if they cannot locate a recipient to accept the message.

The GCCS AMHS is an extension to the AUTODIN messaging system utilizing knowledge based tools to accelerate the delivery processes and procedures while maintaining the same level of delivery integrity as AUTODIN with fewer people. Knowledge base systems are only as good as the expertise of the support people at knowledge engineering, incorporating the rules and configuring the system. The “dead letter” feature described in Section 3 routes all messages not delivered by the rules (Topic profiles) to the system operator for manual routing and delivery. As the knowledge base improves the number of undelivered dead letters should approach zero. The AMHS authoring and release tools also ensure outbound messages are handed off to the AUTODIN system in deliverable format.

As with all previous messaging systems the weak link in the chain is the recipient. The recipient must review incoming messages in a timely manner and accept accountability to act on the information, reject the message or forward it to someone with the authority to act. The GCCS desktop Notify / Alarm lights can be configured for the Executive Manager (EM) Desktop to alert the user of important incoming messages only for coordination and forwarding. The integrity of the messaging system is guaranteed by the Standard Automated Terminal (SAT)/Communications Support Processor (CSP) Backside Terminal (CBT) notifying the switching center that the message has been accepted only after the message has been written into the AMHS file system for routing. The system operator monitors the health of the AMHS and scans the message database daily, checking for anomalous, undelivered messages. Every message traverses from human hand to human hand.

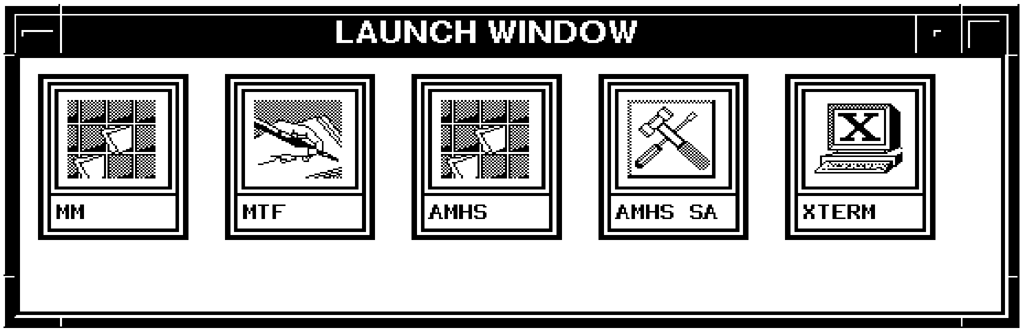
The GCCS AMHS, in its simplest form, is a system that performs three basic functions:

(1) Accepts messages via the SAT/CBT from the AUTODIN Switching Center (ASC), stores them in a Network File System (NFS)-mounted drive on the AMHS Server and routes them to the appropriate recipients based on predetermined profiles and criteria (TOPIC).

(2) Assists users in authoring messages through Message Text Format (MTF) Editor, then routing them for approval through Message Manager (MM). It releases messages, with format validation, to the AUTODIN switch via the Releaser software in Message Manager and SAT/CBT, and processes and stores the comeback copy (CBC).

(3) Searches for key words in a message database using the TOPIC retrospective search tools to locate any message that relates to user-defined criteria. This can be both address and/or content.

AMHS users can execute the Topic Client, Message Manager and MTF Editor applications by using the AMHS, MM, and MTF launch icons, respectively. These icons are located on the desktop application launch window. Access to these icons is granted to users by their GCCS System Administrator using the Profile Manager System Administration Tool otherwise known as the Sys Admin Tool, AMHS Administrators should work with their GCCS System Administrators to ensure the proper users have access to the applications that are required to read, prepare and validate AUTODIN messages. Following is an illustration of these icons.



The primary message processing required by the users is the automatic dissemination of text messages as they are received from AUTODIN. This dissemination is based on plain language address (PLA), office symbol or user's areas of interest (AOIs) as specified by a profile associated with the users uniquely or by group. This profile will contain information on the content of the message (message type, originators, subjects, exercise name, keywords, specific action and information addressees, etc.) to be used to direct messages to corresponding user accounts. The configurable Discretionary Access Controls (DACs) ensure classified messages are only routed to users with pre-authorization even if the message meets the previously defined criteria.

In addition to the receipt of inbound AUTODIN message traffic, the users need tools to assist in the preparation of MIL-STD-DD173, ACP 126 or JANAP 128 text messages, and coordination and release of messages out through the AUTODIN message delivery system. These tools are included in the MTF Editor and Message Manager. Retrospective search of on-line AUTODIN messages on demand, both incoming and archived, greatly assists the users when changing requirements necessitate reviewing recent communiqués. The integration of messaging with the desktop foldering feature assists in electronic filing and suspense tracking.

In this section we will use a layered presentation to develop an understanding of the basic functionality and operation of a GCCS AMHS environment. In Section 3 we will demonstrate a step-by-step process of configuring a typical GCCS AMHS, and in subsequent sections we will address specific operator and administrator tasks.

Section 1 (Introduction) touched on the basic philosophy and architecture of the GCCS. In the site document library should be current documents that describe GCCS in more detail. It is beyond the scope of this document to cover all of these issues, and you are encouraged to become familiar with these reference documents to understand how the AMHS integrates into, and operates in concert with the GCCS to fulfill its mission.

2.1 AMHS ENVIRONMENT

The AMHS operates on a distributed UNIX (Sun/Solaris) LAN-based architecture, and a clear understanding of the hardware/software/UNIX/Solaris environment is mandatory. The site document library will have this reference material. The importance of site-specific documents such as Standard Operating Procedures (SOPs), system and operator logs, and roles and responsibilities are outlined in Section 3.

2.1.1 Sun/Solaris/UNIX Environment

The Sun/Solaris/UNIX environment is a set of hardware and software building blocks that can be used to build a cost effective and flexible system to host the GCCS AMHS. As shown in Figure 2-1, the system is made up of two primary functional blocks, the GCCS Data Center and the User Workstation Community. The Data Center is in a centralized, controlled area and the User Workstations are connected via the GCCS LAN, secure dial-up remote access lines, and other interconnections as appropriate.

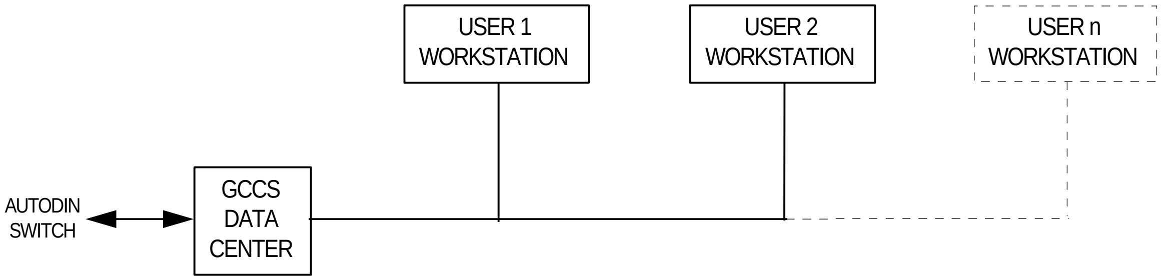


Figure 2-1. AMHS Functional Block Diagram

The user workstations are typically Sun SPARC 5s or HP715s, and in some cases, specially configured PCs. Each workstation has system and client software on a local hard drive and application-specific software that may be soft linked to other drive partitions on other machines. User printers may be local or networked for sharing.

The GCCS Data Center theoretically can be assembled and configured out of anything from a single SPARC to a complex array of SPARC 5s, 10s, 20s, 1000s, and/or 2000s, and accessories such as hard drives, Redundant Arrays of Inexpensive Drives (RAID), tape drives, printers, and other peripherals as required.

The UNIX operating system supports multiprocessing and multitasking and will automatically assign processes to central processing units (CPUs) as necessary to maximize throughput. Most Sun SPARC chassis are designed to support multiple CPUs, which may be added as required to meet system demands. Figure 2-2 shows two chassis with three CPUs each and a typical software stack. Each chassis shows a monitor and keyboard. This is typically known as the console. The console is used to start, stop and monitor UNIX processes.

The underlying strength of the UNIX environment is the inherent security, flexibility and power of the file system. Each directory, executable program and data element is accessible only if appropriate permissions exist.

Permission Field	# of Links	File's Owner	File's Group	Size in Bytes	Date of Last Modification	Filename
drwxrwxrwx	4	E6JSF	staff	1024	Nov 12 12:03	Manpower

The Permissions field is 10 char; 1) -/d=file/directory, 234) user, 567) group, 8910) others, r=read,w=write,x=exec

The concept of permissions (rwx) for files and directories issued to users (owner) and groups of users is heavily utilized by the AMHS. Permissions can be managed by tools within the Executive Manager, Security Manager and Profile Manager, i.e., through Project/Position Pairs. Each physical device, e.g. hard drive or tape drive, has an ID number. The UNIX file system is set up on a partition and each partition has a name. A drive may have multiple partitions or a partition may span multiple drives. The UNIX file system is hierarchical, meaning that the highest name is the root directory labeled **/root**, and under **/root** are subdirectories such as **/h**, **/home**, **/AMHS**, etc., leading down to specific files of interest. The file system is made available to the host by UNIX File System (UFS) mounting the partition on the local hard drive, and partitions on other drives can be made available by NFS mounts (remote mounts). Files within one directory may appear to be in another directory by use of soft links.

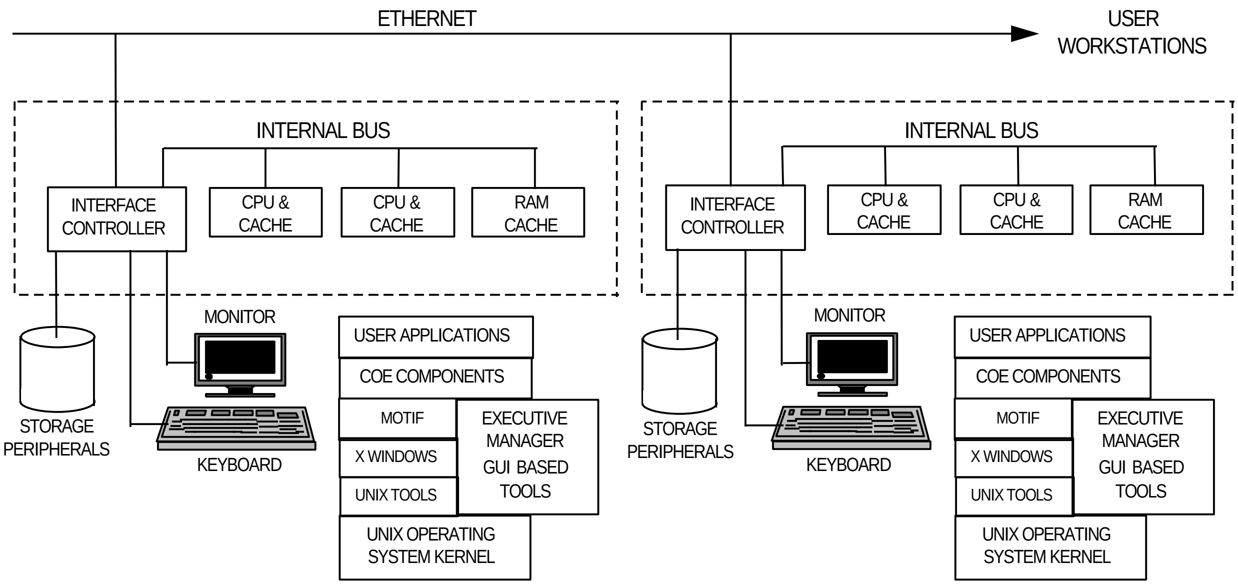
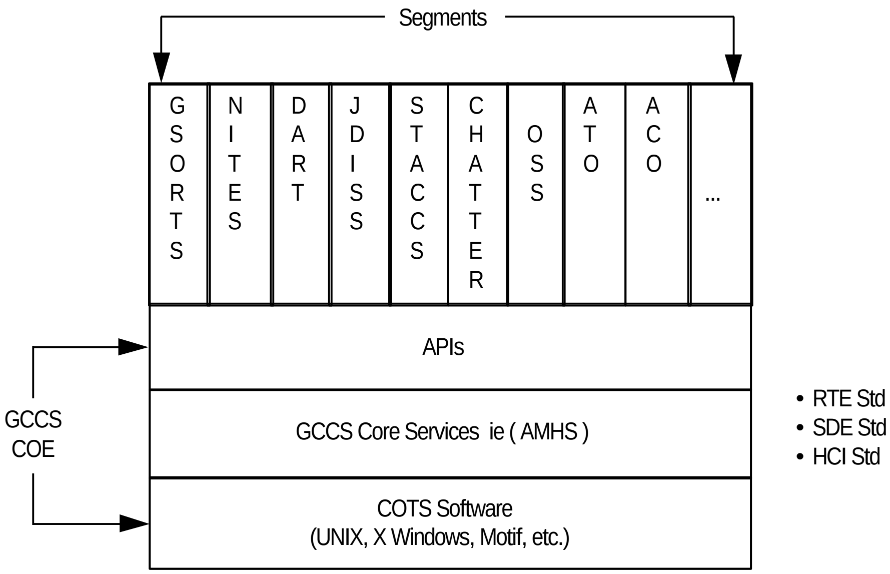


Figure 2-2. UNIX Environment

Host names, Internet Protocol (IP) addresses, Domain names, user names, group names and passwords together with the Read Only Memory (ROM) boot feature all work together with NIS+ and auditing to orchestrate system security and functionality. UNIX operating system commands and tools such as **ps** ,**csh**, **vi, cd** and **ls**, accessible from the Xterm window, assist the operators and administrators. There is a special set of permissions called the superuser (**su**), also known as root privileges. This level of permissions is required for system backup and restores, add and remove users and passwords, add and delete groups and several other special activities. When you use the Xterm **su** command you will be prompted for the su password for root privileges. This login should be avoided whenever possible, and accounts such as **secman**, **sysadmin** and **amhs_dba**, with their appropriate tools, should be used to allow for audit trails and better system controls.

Figure 2-3 shows how user applications connect through APIs to GCCS COE.

This is not an attempt to be a UNIX tutorial but rather a metric for your level of understanding of the UNIX environment. If any of these concepts are unclear, this should assist you in focusing your review of the most relevant concepts.



- Concept is a software ‘backplane”with ‘plug & play”modules.
- Variants determine which segments and how much of core is loaded.
- Standards indicate how to configure segments to use core services.
- APIs are interface between segments and core services.

Figure 2-3. GCCS Software Components

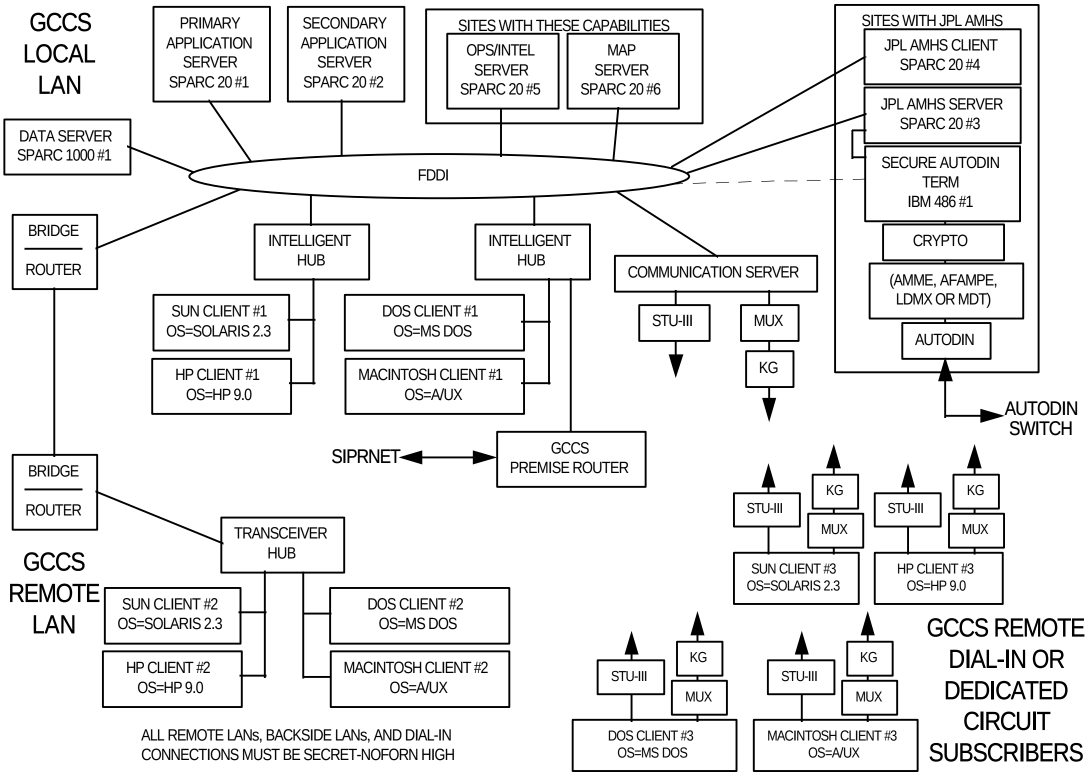

GCCS
LOCAL
LAN
PRIMARY APPLICATION SERVER SPARC 20 #1
SECONDARY APPLICATION SERVER SPARC 20 #2
SITES WITH THESE CAPABILITIES
OPS/INTEL SERVER SPARC 20 #5
MAP SERVER SPARC 20 #6
SITES WITH JPL AMHS
JPL AMHS CLIENT SPARC 20 #4
JPL AMHS SERVER SPARC 20 #3
SECURE AUTODIN TERM IBM 486 #1
CRYPTO
(AMME, AFAMPE, LDMX OR MDT)
AUTODIN
AUTODIN SWITCH
DATA SERVER SPARC 1000 #1
FDDI
BRIDGE
ROUTER
INTELLIGENT HUB
INTELLIGENT HUB
COMMUNICATION SERVER
STU-III
MUX
KG
SUN CLIENT #1 OS=SOLARIS 2.3
HP CLIENT #1 OS=HP 9.0
DOS CLIENT #1 OS=MS DOS
MACINTOSH CLIENT #1 OS=A/UX
SIPRNET
GCCS PREMISE ROUTER
BRIDGE
ROUTER
TRANSCEIVER HUB
GCCS
REMOTE
LAN
SUN CLIENT #2 OS=SOLARIS 2.3
HP CLIENT #2 OS=HP 9.0
DOS CLIENT #2 OS=MS DOS
MACINTOSH CLIENT #2 OS=A/UX
ALL REMOTE LANs, BACKSIDE LANs, AND DIAL-IN CONNECTIONS MUST BE SECRET-NOFORN HIGH
STU-III
KG
MUX
SUN CLIENT #3 OS=SOLARIS 2.3
STU-III
KG
MUX
HP CLIENT #3 OS=HP 9.0
STU-III
KG
MUX
DOS CLIENT #3 OS=MS DOS
STU-III
KG
MUX
MACINTOSH CLIENT #3 OS=A/UX
GCCS REMOTE
DIAL-IN OR
DEDICATED
CIRCUIT
SUBSCRIBERS

Figure 2-4. GCCS Site Hardware Concept

Figure 2-4 shows how hardware might be utilized and interconnected, and specifically how the AMHS component fits into the GCCS hardware environment.

Figures 2-5 and 2-6 show typical small and large site configurations and probably do not reflect how your site is configured. Refer to your AMHS Site Description document for details. These details such as hardware, software and mass storage information should be included in the site AMHS SOP document for cross-reference to this manual.

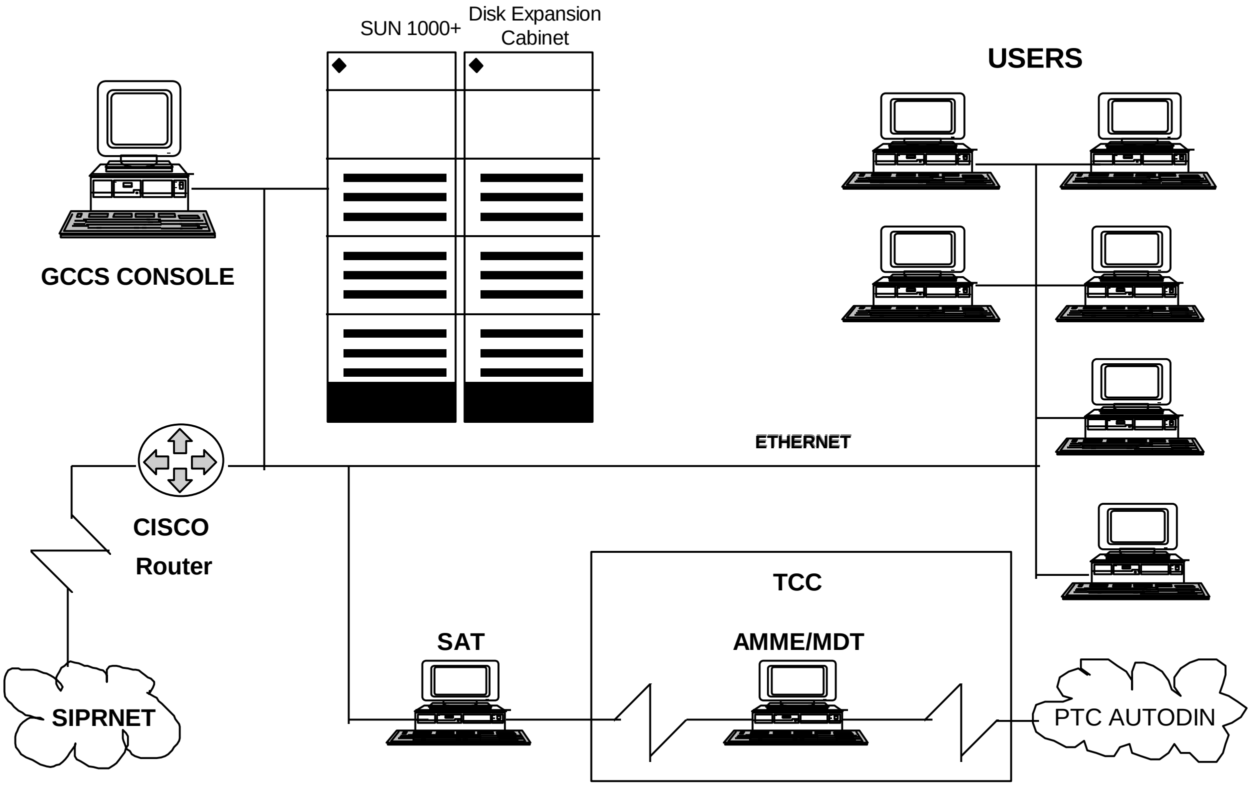


Figure 2-5. Typical Small GCCS AMHS Site

In the typical small site all of the COE components and Applications including AMHS run on a single server and can support a small user community with a low AUTODIN traffic load.

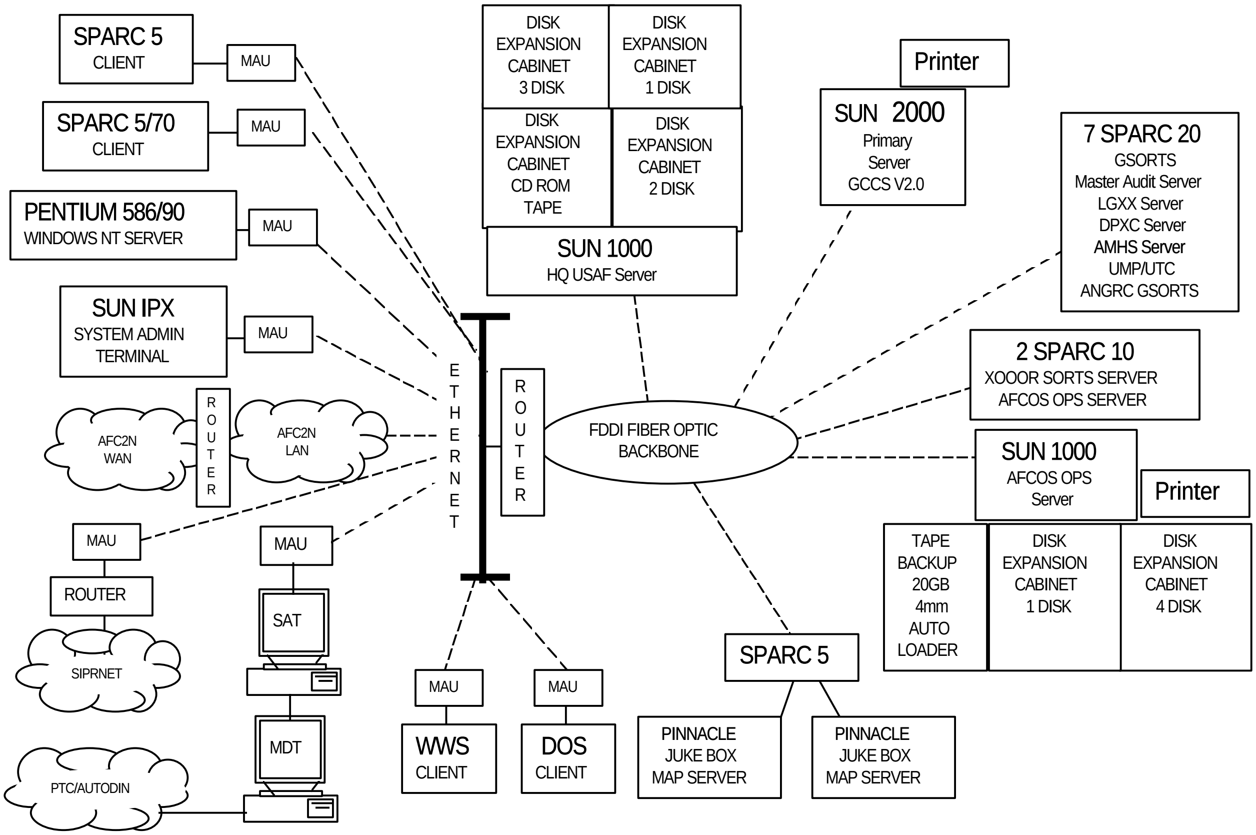


Figure 2-6. Typical Large GCCS AMHS Site

As you can see from the preceding examples the GCCS AMHS environment can vary significantly from installation to installation. The Sun/SPARC/UNIX environment creates a special set of challenges for the system/subsystem administrators and operators in that where the daemons, threads, processes and sessions are running and where the applications and data are stored is site-specific.

2.1.2 AMHS Baseline Environment

For the purpose of this discussion the GCCS/AMHS environment will have four primary hardware components: 1) SPARC 20 EM Server; 2) SPARC 20 AMHS Server, 3) PC-x86 SAT/CBT, and 4) User Workstations with typical peripherals and function specific software. The EM Server includes a monitor and keyboard, with desktop client software to display status information about the health of the system. The AMHS Server has the same with the addition of AMHS Client software and Admin Tools to display AMHS status. A typical site will also have an operator workstation for controlling processes, doing backups and other operator-related duties. These components are interconnected via LAN segments appropriate to the system traffic. Figure 2-7 pictures this system with an overview of what software is included on each component for AMHS functionality. The workstation software included on each server is there to facilitate operation and administration.

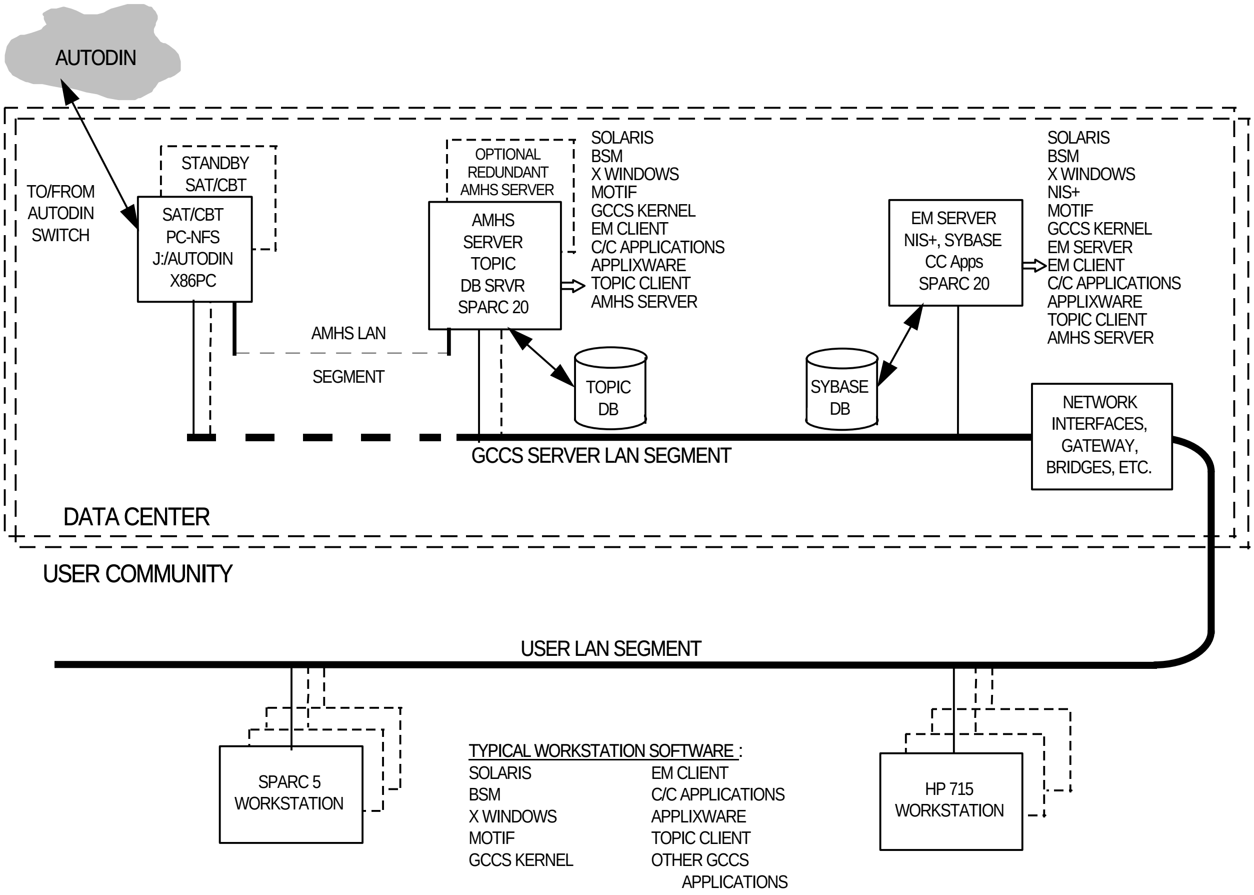


Figure 2-7. AMHS Baseline Environment

2.1.3 SAT/CBT AMHS Server Interface

The PC X86 ISA with the Communications Control Processor II (CCPII) AUTODIN interface card is connected to the Ethernet LAN via an IEEE 802-3 network to the Sun SPARC AMHS Server. The SAT only uses the LAN to read and write to Drive J:\ and can be a dedicated segment to reduce traffic on the GCCS LAN segment. The normal SAT/CBT software runs on Drive C, but in the AMHS system the SAT ini file is modified durring installation to run on drive J:\ and is mounted on Drive J on the Sun SPARC AMHS Server by PC-NFS (Reference Page B-28, Sample SAT.INI, MasterPath = J:\AUTODIN). After initial boot-up, the PC continues all operations on Drive J to facilitate the SAT-AMHS interface. See Figure 2-8.

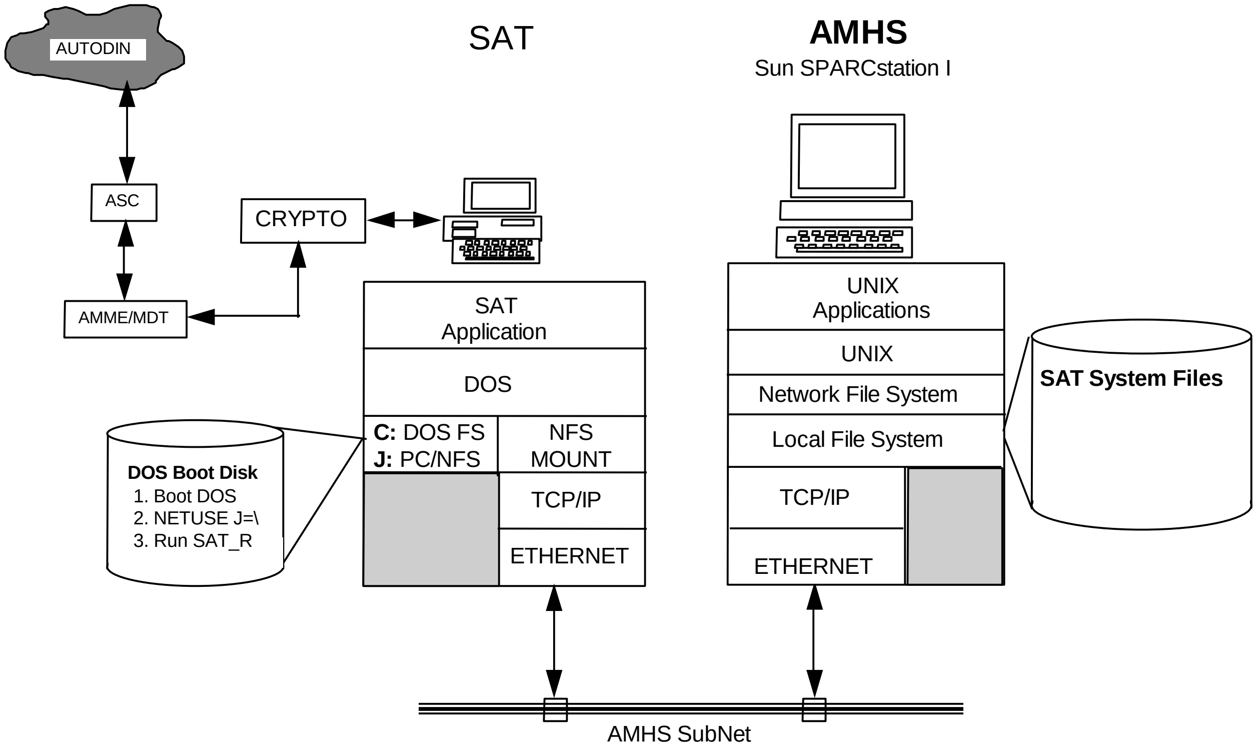
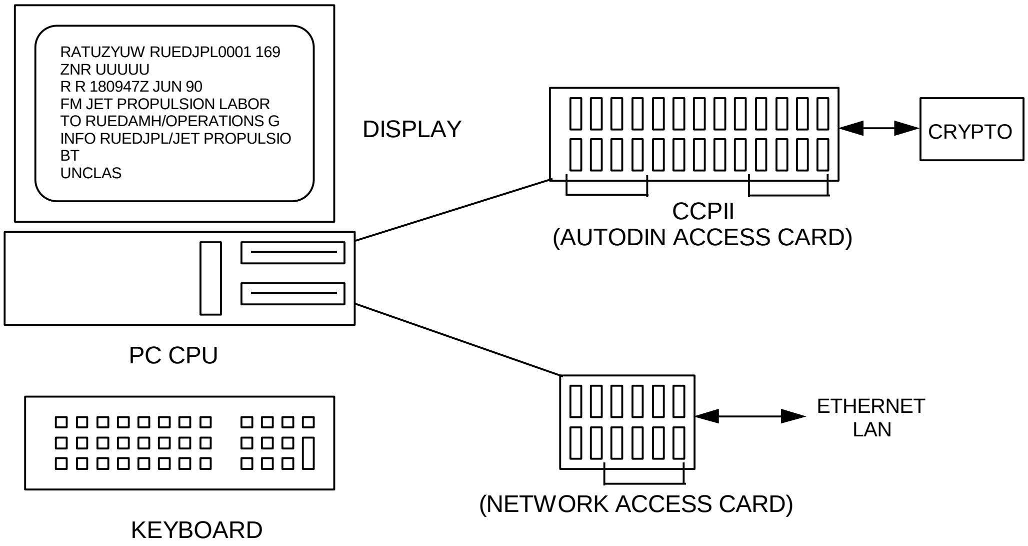


Figure 2-8. Networked SAT/CBT Configuration

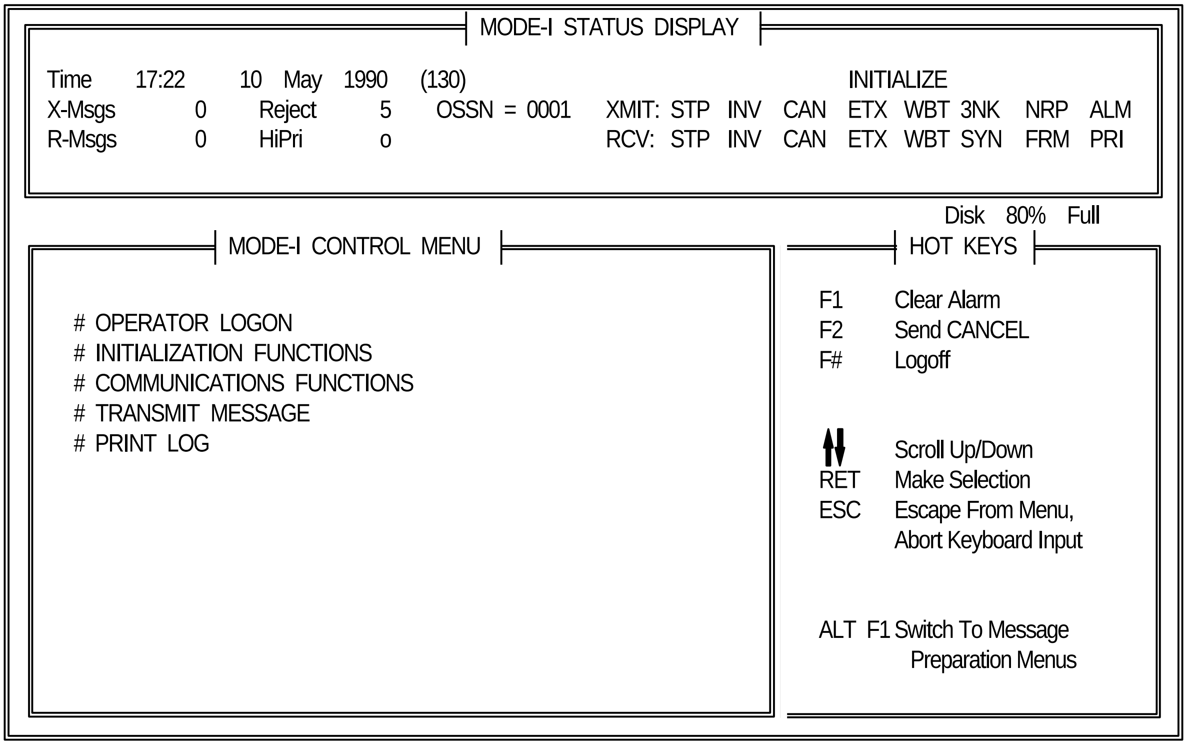
2.1.4 Standard Automated Terminal

The SAT/CBT system from Cavalier Communications Inc. consists of an AUTODIN access card and software for use by a standard PC-compatible 386 or better, with an ISA bus slot. The PC and network card are not part of the Cavalier kit. See Figure 2-9.



```
RATUZYUW RUEDJPL0001 169
ZNR UUUUU
R R 180947Z JUN 90
FM JET PROPULSION LABOR
TO RUEDAMH/OPERATIONS G
INFO RUEDJPL/JET PROPULSIO
BT
UNCLAS
```

Figure 2-9. SAT/CBT Equipment



```
MODE-I STATUS DISPLAY

Time     17:22     10  May  1990   (130)                              INITIALIZE
X-Msgs       0      Reject     5    OSSN = 0001   XMIT: STP  INV  CAN  ETX  WBT 3NK  NRP  ALM
R-Msgs       0      HiPri      o                  RCV:  STP  INV  CAN  ETX  WBT SYN  FRM  PRI

                                                                    Disk  80%  Full
MODE-I CONTROL MENU                                   HOT KEYS

                                                      F1      Clear Alarm
   # OPERATOR LOGON                                   F2      Send CANCEL
   # INITIALIZATION FUNCTIONS                         F#      Logoff
   # COMMUNICATIONS FUNCTIONS
   # TRANSMIT MESSAGE
   # PRINT LOG                                        ⇅       Scroll Up/Down
                                                      RET     Make Selection
                                                      ESC     Escape From Menu,
                                                              Abort Keyboard Input

                                                      ALT F1 Switch To Message
                                                               Preparation Menus
```

Figure 2-10. SAT/CBT Communications Status Display Window

The SAT/CBT display tells the operator the status of the SAT/CBT interface. See Figure 2-10.

2.1.5 Log Maintenance

The SAT/CBT automatically prints the daily log at 00:00:00. These logs should be kept for a minimum of site-specified SOP (60) days. The SAT log lists every file received and transmitted by the SAT, including Format Line 2, file path and file name. See Figure 2-11 for a sample SAT log printout.

```
362 00:00 SAT(CBT) System in Self-Test ( NO ARCHIVAL FILES )
362 10:58 R*** RATTZYUW RUEDJPL0004 0521902-TTTT--RJPL.
              c:\{arch}\r362\121904.004
362 10:58 T*** RATTZYUW RUEDJPL0004 0521902-TTTT--RJPL.
              c:\{arch}\t362\T1000.MSG  <--  T1000.MSG
362 10:58 R*** RATUZYUW RUEDJPL0004 0521902-UUUU--RJPL.
              c:\{arch}\r362\121904AA.004
362 10:58 T*** RATUZYUW RUEDJPL0004 0521902-UUUU--RJPL.
              c:\{arch}\t362\T1001.MSG  <--  T1001.MSG
362 10:58 R*** RATUZYUW RUEDJPL0004 0521902-UUUU--RJPL.
              c:\{arch}\r362\121904AB.004
```

Figure 2-11. SAT/CBT Log File Sample

Table 2-1 describes the software segments installed on each component. The GCCS segment installation document CM #LL-500-103-17 describes in detail how to install GCCS COE and AMHS-specific segments, and your AMHS Site Description document describes site-specific configurations. It is very important that these documents are clearly understood to translate these examples into site-specific operations and maintenance. Issues that need to be understood include: 1) where the segments are installed, 2) what partitions are on which hard drive, 3) what file directory soft links and NFS mounts have been established, and 4) on which CPU the processes and sessions will be running. All this information should be included in the AMHS configuration section of the site SOP document.

Table 2-1. GCCS AMHS Baseline Software Components

Segment Name As of 5 Sep 95	**EM Server**	**AMHS Server**	**GCCS/AMHS Workstation**	**AMHS Requirement(s)**
UNIX OS	X	X	X	
X Windows	X	X	X	
Motif	X	X	X	
BSM*	X	X	X	Loading audit config. files, scripts for loading and reviewing audit logs.
NIS+	X	X	X	Required for AMHS account group/user naming.
Audit	X			Security Auditing
Exec Mgr*	X	X	X	Basic Monitor and Control configuration, needed for profiles.
Sybase*	X			Support for CCAPPS (foldering system).
CCAPPS*	X	X	X	Message Manager and macros for mtf_editor.
AMHS Server(1)		X		AMHS support software for TOPIC server.
AMHS Client	optional	X	X	AMHS support software for TOPIC interaction by client.
TOPIC COTS	optional	X	X	COTS TOPIC product used as text profiler and retrieval engine.
Applix*	X	X	X	COTS Applix product used as basis for mtf_editor, document conversion.
EM Printer	X	X	X	Support for network printing.

***** Indicates GCCS COE/COE Applications Segments

(1) The SAT/CBT Ver 4.10 b/c S/W is included on the segment. PCNFS is also required.

NOTE: See CM #LL-500-103-17 for the latest list and version numbers.

2.1.6 System and Applications Processes

The AMHS Server is running a number of simultaneous processes. These processes can be started and stopped as required both automatically and manually. Table 2-2 shows a typical list of both AMHS-specific and Operating System processes. This listing is generated by typing **ps -eaf**.

Table 2-2. Typical System and Application Processes

```
UID   PID  PPID  C    STIME TTY        TIME COMD
   root     0     0 58   Jan 08 ?         0:01 sched
   root     1     0 80   Jan 08 ?         1:40 /etc/init -r
   root     2     0 80   Jan 08 ?         0:11 pageout
   root     3     0 80   Jan 08 ?        11:48 fsflush
   root   442     1 80   Jan 08 ?         0:01 /usr/lib/saf/sac -t 300
   root 25597   464 73 18:20:01 ?         0:01 /h/EM/progs/msql/bin/msqld
   root   280     1 80   Jan 08 ?         0:36 /usr/sbin/rpcbind
   root   299     1160   Jan 08 ?         0:02 /usr/sbin/inetd -s
   root   290     1 11   Jan 08 ?         0:00 /usr/sbin/kerbd
   root   282     1 80   Jan 08 ?         0:16 /usr/sbin/keyserv
   root   367     1 80   Jan 08 ?         0:01 /usr/lib/sendmail -bd -q1h
   root   288     1 80   Jan 08 ?         0:01 /usr/sbin/nis_cachemgr
   root   306     1 12   Jan 08 ?         0:00 /usr/lib/autofs/automountd
   root   310     1 16   Jan 08 ?         0:00 /usr/lib/nfs/statd
   root   312     1 80   Jan 08 ?         0:03 /usr/lib/nfs/lockd
   root   330     1 80   Jan 08 ?         0:02 /usr/sbin/syslogd
   root 19262 19256 61 16:16:47 pts/9     0:00 csh
   root   340     1 80   Jan 08 ?         0:13 /usr/sbin/cron
   root   429     1 46   Jan 08 ?         0:00 /usr/lib/nfs/mountd
   root   427     1 11   Jan 08 ?         0:00 /usr/lib/nfs/nfsd -a 16
   root   447     1 36   Jan 08 ?         0:00 /h/AcctGrps/SecAdm/progs/AlertDaemon
   root   375     1 80   Jan 08 ?         0:09 /usr/lib/utmpd
   root   358     1 23   Jan 08 ?         0:00 /usr/lib/lpsched
   root   366   358 17   Jan 08 ?         0:00 lpNet
   root   463     1 28   Jan 08 ?         0:00 /h/CCAPPS/progs/map_daemon
   root   444   442 80   Jan 08 ?         0:02 /usr/lib/saf/ttymon
   root   451     1 80   Jan 08 ?         6:55 /h/EM/progs/uccs_local_executive
   root   446     1226   Jan 08 ?         0:00 /h/EM/progs/xdm -config /h/EM/libs/xdm/xdm-config
   root 25589   446 80 18:19:54 ?         0:44 /usr/bin/X11/X -nobanner -auth /usr/lib/X11/xdm/A:0-a0006y
   root 25590   446 80 18:19:55 ?         0:00 /h/EM/progs/xdm -config /h/EM/libs/xdm/xdm-config
   root   464     1170   Jan 08 ?         0:00 /bin/sh /h/EM/progs/msql/bin/run_daemon.edss msqld
   root   468   463 25   Jan 08 ?         0:00 comm_queuer u6mapdmn
  willie 11033     1221   Jan 10 ?        0:01 sm_launcher  20  19
amhs_dba 28334 28309209 18:57:36 pts/1    0:01 -csh
  willie  9614     1 76   Jan 10 ?        0:01 /home1/COTS/APPLIX/axdata/axnet 300176 -fork
amhs_dba 21568     1 80 17:51:46 ?        0:02 rt server -PROCNAME server -_gmtoff 0
   root 25873   299 39 18:25:49 ?         0:00 in.rshd
amhs_dba 21593     1 80 17:51:49 ?        0:02 rt prof -PROCNAME pf1 -_gmtoff 0
  willie 25663 25620 17 18:20:47 ?        0:00 /bin/csh -f /h/AcctGrps/GCCS/Scripts/RunGCCS
  willie 29245 25792 80 19:01:04 ?        0:00 /bin/sh /h/AMHS/Client/progs/amhs_exec_1
   root 19223 19216 77 16:14:59 pts/7     0:00 csh
amhs_dba 21590     1 80 17:51:48 ?        0:01 rt prof -PROCNAME pf0 -_gmtoff 0
amhs_dba   551 28334  5 19:21:32 pts/1    0:01 /bin/sh topic_cmd
amhs_dba  1240     1 59 19:26:56 pts/1    0:00 cbc_feed
  willie 25620 25590132 18:20:32 ?        0:01 /bin/csh /h/USERS/willie/Scripts/.xsession
amhs_dba  1209     1 50 19:26:52 pts/1    0:00 sat_feed
   root 19208 19201 61 16:14:30 pts/6     0:00 csh
   root   860   299 18   Jan 08 ?         0:00 rpc.rstatd
   root 19199   299 77 16:14:21 ?         0:01 in.rlogind
   root 19214   299 80 16:14:45 ?         0:01 in.rlogind
   root 19163 19161 80 16:12:55 pts/2     0:00 -sh
   root 12407   299 80 01:19:58 ?         0:05 in.rlogind
   root 19216 19214 73 16:14:46 pts/7     0:00 -sh
   root 28309 28300 64 18:57:13 pts/1     0:00 sh
   root 19235   299 66 16:16:09 ?         0:01 in.rlogind
   root  6531     1 80   Jan 09 ?         0:00 csh
amhs_dba  1203     1 80 19:26:52 pts/1    0:01 rt merge -PROCNAME mg1 -_gmtoff 0
```

```
  willie 28300 28299 80 18:56:58 pts/1      0:01 csh
amhs_dba  1237     1 68 19:26:55 pts/1      0:01 rt build -PROCNAME dp4 -_gmtoff 0
  willie 29236 29235 80 19:00:51 pts/13     0:01 csh
    root 19416 19410 56 16:44:44 pts/12     0:01 csh
    root 19408   299 80 16:44:26 ?          0:03 in.rlogind
    root 19287 19281236 16:19:10 pts/11     0:01 csh
    root 19279   299 80 16:18:53 ?          0:07 in.rlogind
amhs_dba 25875 25874 80 18:25:50 ?          0:10 /h/AMHS/Server/progs/amhs_admin
  willie 25792 25791 80 18:21:38 ?          0:00 sm_launcher  20  19
    root 19281 19279 72 16:18:54 pts/11     0:00 -sh
    root  6623     1 80   Jan 09 ?          0:04 ttsession -s -d usw16:0.0
    root 19237 19235 66 16:16:09 pts/8      0:00 -sh
  willie 26036 25792 22 18:29:19 ?          0:00 /bin/sh /h/AMHS/Client/progs/start_sa
    root 19184   299 80 16:13:59 ?          0:01 in.rlogind
    root  6624   299 29   Jan 09 ?          0:00 rpc.ttdbserverd
    root 12409 12407 69 01:19:59 pts/10     0:00 -sh
  willie 25791 25668 80 18:21:06 ?          0:24 session_manager -alert_text GCCS Rev 2.1
    root 19161   299 80 16:12:54 ?          0:09 in.rlogind
    root 19243 19237 50 16:16:17 pts/8      0:00 csh
    root 19254   299 70 16:16:36 ?          0:01 in.rlogind
    root 19256 19254 71 16:16:37 pts/9      0:00 -sh
    root 19186 19184 72 16:14:00 pts/4      0:00 -sh
    root 19193 19186 80 16:14:12 pts/4      0:00 csh
  willie 28299 25792 80 18:56:53 ?          0:01 /usr/bin/X11/xterm
    root 19201 19199 75 16:14:21 pts/6      0:00 -sh
    root  1427 19287 48 19:29:19 pts/11     0:00 ps -eaf
  willie 29235 25792 55 19:00:48 ?          0:01 /usr/bin/X11/xterm
    root 12415 12409 80 01:20:05 pts/10     0:01 csh
  willie 25622     1 14 18:20:33 ?          0:00 xconsole -geometry 480x130+1400+100 -daemon -notify -verbose -fn
    root 19410 19408 76 16:44:26 pts/12     0:00 -sh
amhs_dba  1206     1 78 19:26:52 pts/1      0:00 rt build -PROCNAME dp1 -_gmtoff 0
    root  1425 19178  6 19:29:17 pts/2      0:00 ada test_as_.ada
  willie 25661 25620 80 18:20:41 ?          0:18 mwm
    root 26040   299 39 18:29:21 ?          0:00 in.rshd
  willie 25668 25663 80 18:20:47 ?          0:01 /bin/csh /h/AcctGrps/GCCS/progs/GCCSMain
  willie 29315 29245 80 19:01:19 ?          0:07 /h/COTS/Topic/current/bin/xtopic
    root 19178 19163 36 16:13:41 pts/2      0:01 csh
  willie 26039 26036 18 18:29:20 ?          0:00 /usr/ucb/rsh amhserver -l amhs_dba umask 7; setenv LD_LIBRARY
amhs_dba  1234     1 19 19:26:55 pts/1      0:00 rt merge -PROCNAME mg4 -_gmtoff 0
```

2.1.7 TOPIC Database Processes

To be fully operational, the TOPIC database system requires that nine (9) independent processes be up and running. These processes are divided into two types: Servers processes and Real-Time processes. Table 2-3 lists the processes and describes their functions.

Table 2-3. Database Processes

PROCESS NAME		AMHS PROCESSOR
rt server	Database server.	AMHS Server
rt build - dp1	"Document preparation" of inbound AUTODIN messages to load them into the message database.	AMHS Server
rt merge - mg1	Merge process to consolidate inbound AUTODIN messages in the message database.	AMHS Server
rt build - dp4	"Document preparation" of the "comeback" copies of messages to load them into the message database.	AMHS Server
rt merge - mg4	Merge process to consolidate the "comeback" copies of messages in the message database.	AMHS Server
rt prof-pf0	The central profiling process that determines the distribution of messages.	AMHS Server
rt prof-pf1	The central profiling process that determines the distribution of messages.	AMHS Server
rt prof-pfx	The central profiling process that determines the distribution of messages.	AMHS Server
sat_feed	Initial processing of inbound messages received by the Standard Automated Terminal (SAT/CBT).	AMHS Server
cbc_feed	Initial processing of the "comeback" copies of outbound messages.	AMHS Server

NOTE: **pf0** handles the Topic profiling of AUTODIN traffic processed by applications using the API. **pf1** is the default, plain text Topic Profiler, while **pfx** (**pf1**, **pf2**,...**pfx**) are additional profilers that may be added when necessary for performance tuning. An example would be when the server mailbox (Figure 2-22) is nearly full, or the system is very slow. Table 2-4 shows typical server process running. Table 2-5 shows typical user process running.

Table 2-4. Typical Database Processes

```
amhs_dba 28334 28309209 18:57:36 pts/1      0:01 -csh
amhs_dba 21568     1 80 17:51:46 ?          0:02 rt server -PROCNAME
server -_gmtoff 0
amhs_dba 21593     1 80 17:51:49 ?          0:02 rt prof -PROCNAME pf1 -
_gmtoff 0
amhs_dba 21590     1 80 17:51:48 ?          0:01 rt prof -PROCNAME pf0 -
_gmtoff 0
amhs_dba   551 28334  5 19:21:32 pts/1      0:01 /bin/sh topic_cmd
amhs_dba  1240     1 59 19:26:56 pts/1      0:00 cbc_feed
amhs_dba  1209     1 50 19:26:52 pts/1      0:00 sat_feed
amhs_dba  1203     1 80 19:26:52 pts/1      0:01 rt merge -PROCNAME
mg1 -_gmtoff 0
amhs_dba  1237     1 68 19:26:55 pts/1      0:01 rt build -PROCNAME dp4
-_gmtoff 0
amhs_dba 25875 25874 80 18:25:50 ?          0:10
```

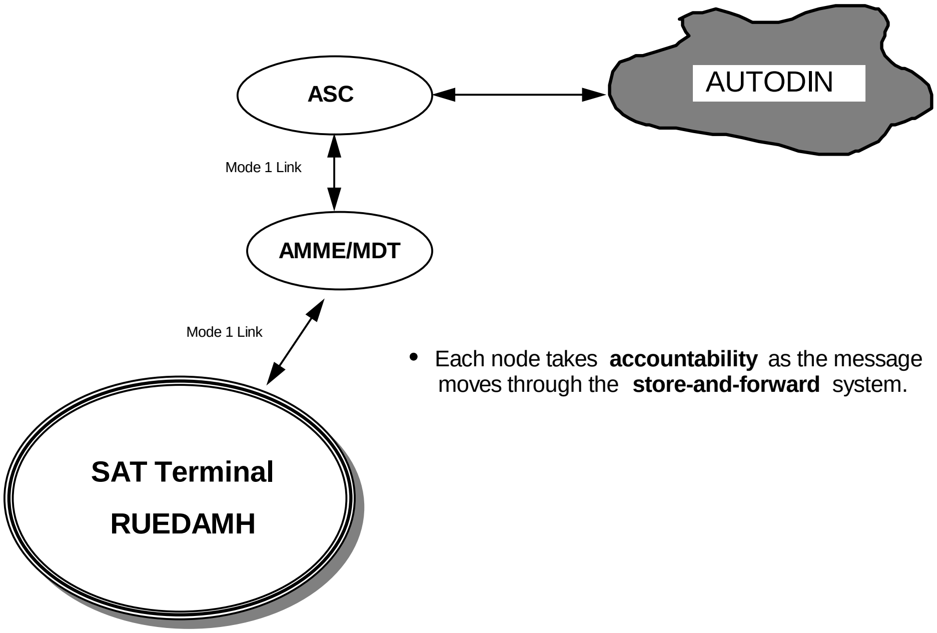
```
/h/AMHS/Server/progs/amhs_admin
amhs_dba  1206     1 78 19:26:52 pts/1      0:00 rt build -PROCNAME dp1
-_gmtoff 0
amhs_dba  1234     1 19 19:26:55 pts/1      0:00 rt merge -PROCNAME
mg4 -_gmtoff 0
```

Table 2-5. Typical User Processes

```
willie  9614     1 76   Jan 10 ?        0:01 /home1/COTS/APPLIX/axdata/axnet 300176 -fork
willie 25663 25620 17 18:20:47 ?        0:00 /bin/csh -f
/h/AcctGrps/GCCS/Scripts/RunGCCS
willie 29245 25792 80 19:01:04 ?        0:00 /bin/sh /h/AMHS/Client/progs/amhs_exec_1
willie 25620 25590132 18:20:32 ?        0:01 /bin/csh /h/USERS/willie/Scripts/.xsession
willie 28300 28299 80 18:56:58 pts/1    0:01 csh
willie 29236 29235 80 19:00:51 pts/13   0:01 csh
willie 25792 25791 80 18:21:38 ?        0:00 sm_launcher  20  19
willie 26036 25792 22 18:29:19 ?        0:00 /bin/sh /h/AMHS/Client/progs/start_sa
willie 25791 25668 80 18:21:06 ?        0:24 session_manager -alert_text GCCS Rev 2.1
willie 28299 25792 80 18:56:53 ?        0:01 /usr/bin/X11/xterm
willie 29235 25792 55 19:00:48 ?        0:01 /usr/bin/X11/xterm
willie 25622     1 14 18:20:33 ?        0:00 xconsole -geometry 480x130+1400+100 -
daemon -notify -verbose -fn
willie 25661 25620 80 18:20:41 ?        0:18 mwm
willie 25668 25663 80 18:20:47 ?        0:01 /bin/csh /h/AcctGrps/GCCS/progs/GCCSMain
willie 29315 29245 80 19:01:19 ?        0:07 /h/COTS/Topic/current/bin/xtopic
willie 26039 26036 18 18:29:20 ?        0:00 /usr/ucb/rsh amhserver -l amhs_dba umask 7;
setenv LD_LIBRARY_PATH
```

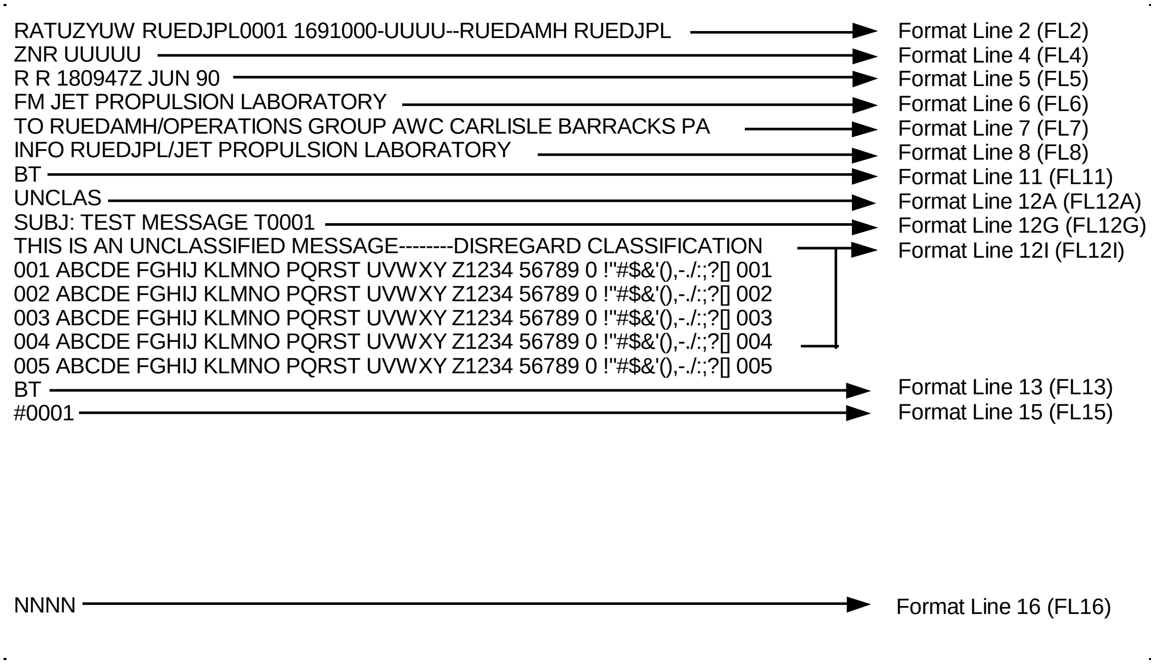
2.2 DEFENSE COMMUNICATION SYSTEM (DCS)

The DCS AUTODIN system is a worldwide Department of Defense computerized general purpose communications system which provides for the transmission of narrative and data pattern traffic on a store-and-forward (message switching) basis. JANAP 128 is the primary format. The SAT/CBT is connected to the nearest AUTODIN Switch with a synchronous data communications link. Figure 2-12 shows the AUTODIN link configuration and Figure 2-13 illustrates a typical message.



- Each node takes **accountability** as the message moves through the **store-and-forward** system.

Figure 2-12. AUTODIN Link Configuration

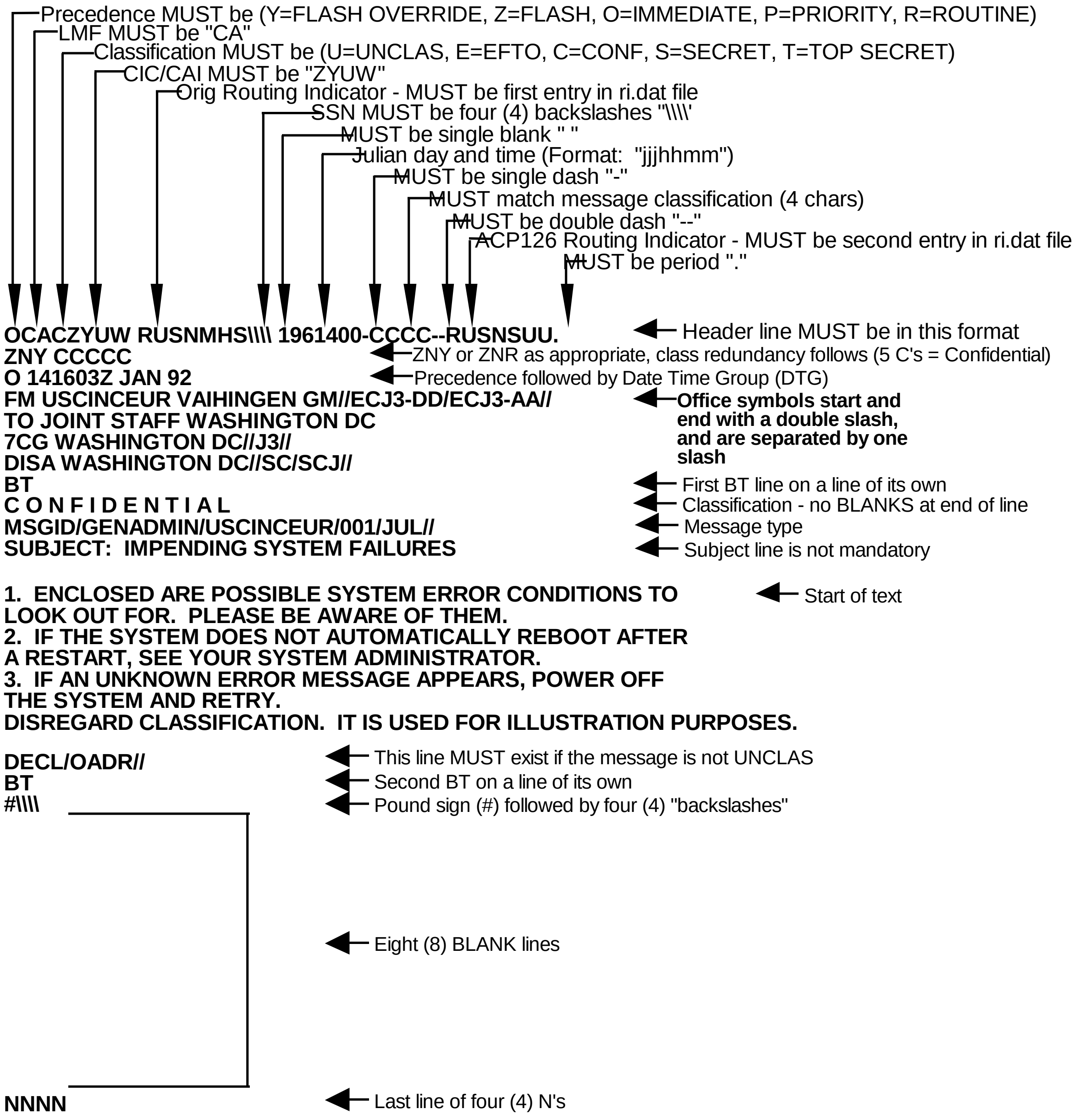


```
RATUZYUW RUEDJPL0001 1691000-UUUU--RUEDAMH RUEDJPL
ZNR UUUUU
R R 180947Z JUN 90
FM JET PROPULSION LABORATORY
TO RUEDAMH/OPERATIONS GROUP AWC CARLISLE BARRACKS PA
INFO RUEDJPL/JET PROPULSION LABORATORY
BT
UNCLAS
SUBJ: TEST MESSAGE T0001
THIS IS AN UNCLASSIFIED MESSAGE--------DISREGARD CLASSIFICATION
001 ABCDE FGHIJ KLMNO PQRST UVWXY Z1234 56789 0 !"#$&'(),-./:;?[] 001
002 ABCDE FGHIJ KLMNO PQRST UVWXY Z1234 56789 0 !"#$&'(),-./:;?[] 002
003 ABCDE FGHIJ KLMNO PQRST UVWXY Z1234 56789 0 !"#$&'(),-./:;?[] 003
004 ABCDE FGHIJ KLMNO PQRST UVWXY Z1234 56789 0 !"#$&'(),-./:;?[] 004
005 ABCDE FGHIJ KLMNO PQRST UVWXY Z1234 56789 0 !"#$&'(),-./:;?[] 005
BT
#0001




NNNN
```

Figure 2-13. Typical Message

2.2.1 Standard AUTODIN Message Types

The SAT/CBT stores the incoming message in raw format and the AMHS processes through messages in DD173 format. As shown in Figure 2-19, the raw messages are processed by the **sat_feed** process and stored in the DAC DIR. Several lines have periods added to the beginning of the line as defined by the **add_dot** statements in the **vardef** file. The \\\\'s in this format are station serial number (SSN) IDs and are controlled by the SAT. For example, in Figure 2-13 the \\\\'s were replaced with 0001 by the SAT. Below may be found the anatomy of a DD173 message, Figure 2-14, consistent with JANAP 128 message format, followed by samples of DD173 (Figure 2-15), JANAP 128 (Figure 2-16), and ACP 126 (Figure 2-17) messages.



Precedence MUST be (Y=FLASH OVERRIDE, Z=FLASH, O=IMMEDIATE, P=PRIORITY, R=ROUTINE)
LMF MUST be "CA"
Classification MUST be (U=UNCLAS, E=EFTO, C=CONF, S=SECRET, T=TOP SECRET)
CIC/CAI MUST be "ZYUW"
Orig Routing Indicator - MUST be first entry in ri.dat file
SSN MUST be four (4) backslashes "\\\\'
MUST be single blank " "
Julian day and time (Format: "jjjhhmm")
MUST be single dash "-"
MUST match message classification (4 chars)
MUST be double dash "--"
ACP126 Routing Indicator - MUST be second entry in ri.dat file
MUST be period "."

```
OCACZYUW RUSNMHS\\\\ 1961400-CCCC--RUSNSUU.
ZNY CCCCC
O 141603Z JAN 92
FM USCINCEUR VAIHINGEN GM//ECJ3-DD/ECJ3-AA//
TO JOINT STAFF WASHINGTON DC
7CG WASHINGTON DC//J3//
DISA WASHINGTON DC//SC/SCJ//
BT
C O N F I D E N T I A L
MSGID/GENADMIN/USCINCEUR/001/JUL//
SUBJECT:  IMPENDING SYSTEM FAILURES

1.  ENCLOSED ARE POSSIBLE SYSTEM ERROR CONDITIONS TO
LOOK OUT FOR.  PLEASE BE AWARE OF THEM.
2.  IF THE SYSTEM DOES NOT AUTOMATICALLY REBOOT AFTER
A RESTART, SEE YOUR SYSTEM ADMINISTRATOR.
3.  IF AN UNKNOWN ERROR MESSAGE APPEARS, POWER OFF
THE SYSTEM AND RETRY.
DISREGARD CLASSIFICATION.  IT IS USED FOR ILLUSTRATION PURPOSES.

DECL/OADR//
BT
#\\\\








NNNN
```

Header line MUST be in this format
ZNY or ZNR as appropriate, class redundancy follows (5 C's = Confidential)
Precedence followed by Date Time Group (DTG)
Office symbols start and end with a double slash, and are separated by one slash
First BT line on a line of its own
Classification - no BLANKS at end of line
Message type
Subject line is not mandatory
Start of text
This line MUST exist if the message is not UNCLAS
Second BT on a line of its own
Pound sign (#) followed by four (4) "backslashes"
Eight (8) BLANK lines
Last line of four (4) N's

Figure 2-14. Anatomy of an AUTODIN Message

```
01       131700Z  JUN  95  RR  UUUU      AA  ZYUW  RUSNNOA

NO

                 USCINCEUR VAIHINGEN GE

                 1 ISG LINDSEY AS GM //GCCSUSER//

UNCLAS

SUBJ: DD173 - QUICK TEST
THIS IS AN UNCLASSIFIED GCCS TEST MESSAGE, PLEASE DO NOT DISTRIBUTE....
```

Figure 2-15. Sample of a DD173 Message

```
RAAUZYUW RUSNMHS\\\\ 1601636-UUUU--RUSNNOA.
ZNR UUUUU
R 131702Z JUN 95
FM USCINCEUR VAIHINGEN GE
TO 1 SOW HURLBURT FLD FL //GCCSUSER//
BT
UNCLAS
SUBJ: JANAP 128 QUICK TEST

THIS IS AN UNCLASSIFIED GCCS TEST MESSAGE, PLEASE DO NOT DISTRIBUTE....

BT
#\\\\




NNNN
```

Figure 2-16. Sample of a JANAP 128 Message

```
RAAUZYUW RUSNMHS\\\\ 1601636-UUUU--RUSNSUU.
ZNR UUUUU
R 091713Z JUN 95
FM USCINCEUR VAIHINGEN GE
TO 1 SOW HURLBURT FLD FL
BT
UNCLAS
SUBJ:A SHORT QUICK ONE AGAIN

BT
#\\\\





NNNN
```

Figure 2-17. Sample of an ACP 126 Message

In the case of an ACP 126 message that has header line with the Routing Indicator (RI) ending in **SUU**, the system is instructing the Switching Center to do the PLA check and message build. For DD173 and JANAP 128 messages, the PLA lookup and final message build is done by the SAT/CBT. It is important to note that the AMHS releases messages in card format, which will generate a Language Media Format (LMF) of "CA" for card format ASCII text. The default format of the outgoing message is card format. The message format is configurable by changing **Message_Release.ini** settings (see Figure 2-18) in **/h/AMHS/clients/config**. The PROCESS_LMF line forces the output to CA when "= FALSE", or allows the author to define the LMF when "= TRUE".

```
#
# This is the message release configuration
# file. It contains the parameters neccessary
# to customize the message release process.
#

CONVERT_JANAP_128_TO_DD173=TRUE
PROCESS_LMF=[FALSE or TRUE]
```

Figure 2-18. Message_Release.ini

2.3 INBOUND MESSAGE PROCESSING

The AMHS inbound message processing is built upon the Verity Inc. TOPIC Real-Time Database product, which provides AUTODIN message delivery to AMHS Users as well as real-time retrospective search capabilities of the entire System High message database. The TOPIC product stores information about messages in "partitions", subdirectories of message files. A partition can contain information on one or many messages depending on how closely messages are received. Message information is stored in Verity Databases (VDBs) in two parts:

- Document Dataset VDB—stores values for key field names for a message such as precedence, classification, subject, the location of the message, and a pointer to the virtual document definition used to display the message.
- Document Index VDB—stores the full-word inversion for every message in a partition.

A partition is a basic component of the TOPIC system architecture and promotes quick and efficient messages retrievals. Partitions are located in the AMHS Real-Time directory located at **/h/AMHS/Server/topic/amhs_db** on the AMHS Server. Feed processes are used to get the messages to the TOPIC Real-Time processes. AMHS messages are received via two sources or "feeds" called the **sat_feed** and the **cbc_feed**. The **sat_feed** processes inbound JANAP 128 AUTODIN messages and the **cbc_feed** processes the "comeback copy" of a successfully transmitted message. Inbound AUTODIN messages are received at a backside PC terminal hosting Cavalier Communications' SAT/CBT software.

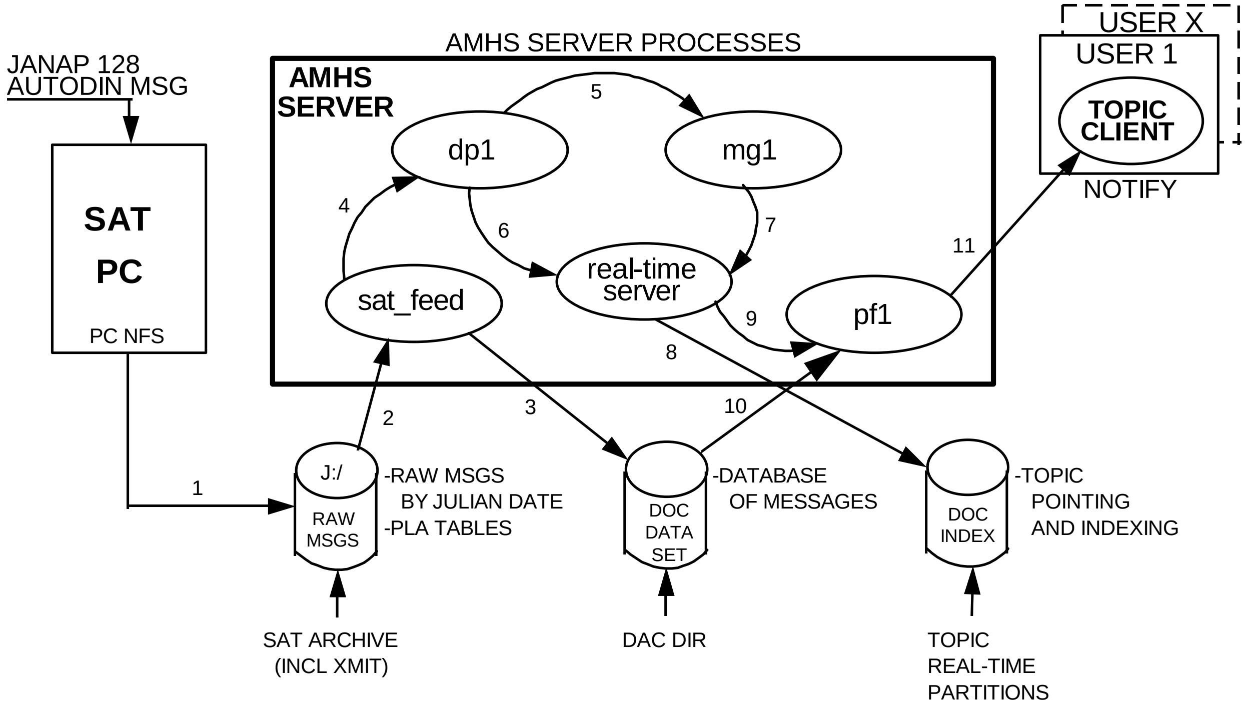


Figure 2-19. AMHS Processes Map

Figure 2-19 identifies the processes in the AMHS that interface to the Real-Time TOPIC product and the AUTODIN backside terminal SAT/CBT process. Each feed has TOPIC database builder (**dp1**) and partition merger (**mg1**) processes associated with it. The database builder (**dp1**) receives incoming messages, assigns document IDs to each message, stores them in temporary partitions (referred to as real-time partitions), then indexes these partitions. Once a partition of messages is created, the database builder sends a notification to the TOPIC Server process which then makes the message available to any AMHS user. The partition merger (**mg1**) receives messages from the database builder when new real-time partitions are available, consolidates the real-time partitions into larger, merged real-time partitions, then notifies the TOPIC Server of their availability.

The TOPIC Server communicates using a real-time interprocess communications via mailboxes which are UNIX file directories.

The Central Profiler (**pf1**) is a TOPIC process which runs independent of the feeds and their BUILD and MERGE processes. There can be several Central Profiler processes; each process has a unique name, e.g. **pf0**, **pf1**, **pf2** .... The Central Profiler receives notification from the TOPIC Server when a new partition of one or more messages is available. The Central Profiler uses a file of user queries (typically called the “profiles file”) that are made up of keywords and/or phrases matching a user’s area of interest. The Central Profiler executes the queries in this file against the partition of one or more messages as it receives notification of a new partition. If a match is made, the AMHS user is notified of the “hit” in one of his special queues configured for that user. This notification is referred to as a “profile hit”.

When messages are “profiled” to a message queue, the AMHS users do not receive “copies” of messages but rather “copies of pointers” to the message in the database. There is only one copy of a message in the TOPIC database. This copy resides in the DAC directories as one file per message.

The AMHS is designed with flexibility and expandability hooks built in. The default configuration for processing SAT-fed AUTODIN messages can easily be expanded with other feeds such as Rueters news services to allow users to receive newswire information consistent with their areas of interest. The system can also easily be expanded to support redundant hardware (dual string AMHS) if the mission dictates this requirement.

To ease AMHS integration with other applications that require AUTODIN interfaces, the system includes an API developers tool set. Currently, the GCCS Reconnaissance Information System (GRIS) application utilizes this feature. (See Appendix B, for details.) The API hooks are described in detail in the GCCS Automated Message Handling System Applications Programming Interface document, Release 1, published by JPL as D-12731.

2.3.1 SAT_FEED

The **sat_feed** gets JANAP 128 AUTODIN messages via the SAT/CBT software and feeds them into the TOPIC Real-Time text retrieval system. The SAT/CBT does the Mode 1 protocol “handshake” with the AUTODIN switch and then creates a message token in one of five precedence ordered backside queues. These queues are referred to as **bsq1-bsq5**, where **bsq1** is for EMERGENCY traffic, **bsq2** is for FLASH traffic, **bsq3** is for IMMEDIATE traffic, **bsq4** is for PRIORITY traffic and **bsq5** is for ROUTINE traffic. The SAT/CBT also copies the raw message into a receive archive directory labeled by an “r” and the Julian date, e.g. r123. Before feeding the message into TOPIC. The **sat_feed** also performs other functions associated with the processing of AUTODIN messages. These other major functions include:

(1) Assignment of Discretionary Access Control (DAC) of messages based on a keyword scan.

(2) Collation of message sections of multisection messages received from AUTODIN.

(3) Delivery record generation for AUTODIN messages.

(4) Message audit functions.

2.3.2 PROFILE_HIT

The **profile_hit** executable is called by the Verity TOPIC Central Profiler process when a message matches a profile registered with the Central Profiler. The calling arguments to the **profile_hit** executable are controlled by the profiles file on which the TOPIC Central Profiler process has focus. These calling arguments contain the recipient’s topic account name and queue name. **profile_hit** updates the message’s delivery record and then takes the appropriate action to ensure delivery of the message to the intended recipient.

2.3.3 CBC_FEED

The **cbc_feed** monitors messages transmitted to AUTODIN by the AMHS and returns a copy of the as-transmitted message, along with all coordination information via a message access record (MAR), to the Verity TOPIC Real-Time system for profile distribution to the appropriate AMHS users.

2.3.4 DACLIST File

The **daclist** file defines a list of code words for each message category type which can be changed to accommodate site-specific needs. The **daclist** file is edited and maintained by the DAC Manager tool. Following is the normal **daclist** file location and contents.

/h/amhs/topic/amhs_db/daclist

```
 Define the list of key words for each message category type.
 This file can be changed to accommodate site specific needs.

 You may use carriage returns and tabs to format this file;
 however, blanks (" ") will be interpreted as such.  Vardef variable
 names may also be used for the DAC information as long as the variables
 are defined in the VARDEF file.  Use of VARDEF variables for the topic
 group and UNIX information fields is strongly discouraged since things
 tend to become confusing operationally.


***** FORMAT OF AN ENTRY *****

First line of entry (must be all on one line):
<Type Name>:<search method>:<dir>:<UNIX group>:<UNIX protection>:
                <topic group>=

Subsequent lines (as many as needed):
  <CODEWORD1>; ...; <CODEWORDn>[@]     (final line is terminated with "@")

***** DESCRIPTION OF FIELDS: *****

Type name - the name of the discretionary access type defined by this
  entry.  This name will appear in the audit trails.

*****

search method - is a number between 1 and 3 as follows

      1 -  Precedence search - use the first dac type that incurs a key
           word hit

      2 -  combination search - "OR" the topic mask for this 'hit' with
           the topic masks from other hits

      A VARDEF variable whose definition is one of these values may also
      be used.

*****

dir - is the name of the sub directory into which messages of this type
 should be placed.  This subdirectory is relative to the directory named
 by the VARDEF variable "dac_dir". The subdirectory may also be specified
 by using a VARDEF variable.

*****
UNIX group - is the UNIX group ownership assigned to messages of this type.
   For UNIX installations, the group named must be a valid UNIX group. For
   non-UNIX installations, this field is not used but must still be valid
   in all other respects. Zero (0) is a reasonable choice for non-UNIX
   installations. (See "Validation applied to all fields")
```

****

UNIX protections - is the UNIX file protection mask assigned to messages of this type. It may consist of three octal digits the last of which must be zero, or else it may consist of a VARDEF variable whose definition meets these criteria and it must be valid in all other respects. For non-UNIX installations, this field is not used but must still be valid. (See "Validation applied to all fields")

****

Topic group - This is the topic security group to which messages of this type are assigned. This field may contain a DECIMAL number which is any one of the following:

- a topic security group number between 0 and 15 or a VARDEF variable whose definition is between 0 and 15.

- the number -1 or a VARDEF variable whose definition is -1. The VARDEF variable "PROTECT" (-1) is typically used. This value indicates that any message of this type is NOT to be fed to Topic.

- the decimal number 16 or a VARDEF variable set to to this value. The VARDEF variable ALL_GROUPS is typically used. This value makes the message accessible to all topic security groups.

****

CODEWORDS - When any of these words or patterns is recognized in the message being searched, then the message is deemed to be of this type. Code words are delimited by semicolons(;). The relationship between the key words listed for a given dac type is an "OR" relation.

A key word may consist of one or more "terms". Each term is an expression upon which a match with the searched text is performed. The relationship between the terms comprising a key word is an "AND" relation. That is, all terms comprising a key word must be matched before a hit on that key word is declared. Terms within a key word are separated by spaces.

The terms of a key word can be specified in one of three ways.

- If a term is enclosed in single forward quotes (') then it is searched for as a "STEM". That is, a match is declared if the sequence of charcters in the specified word is matched in the searched text. In the searched text, any number of whitespace charcters may appear between the characters matched.

- If a term is enclosed in double quotes (") then it is searched for as a "LITERAL". That is, a match is declared if the specified word is matched exactly, as a separate word with no embedded white space.

- If a term is enclosed in back quotes (`) then it is searched for as a "REGULAR EXPRESSION". That is, a match is declared if the specified regular expression is matched. The regular expressions are of the type used by the UNIX utility "ed" with the exception that the "^" and "$" characters designate beginning and end of search string rather than beginning and end of line. The search string is the portion of the message which is searched for key words.

****

Validation applied to all fields.

- The field must be non-null.

- It must not contain any of the special characters ":=;@".

- If it references a VARDEF variable (it begins with "$") then the referenced VARDEF variable must be defined in the VARDEF file.

****

This header may be amended, but the "$eoh" delimiter MUST remain at the end of the header to indicate the beginning of the data.

```
========================================================
$EOH

Cwp:1:$cwp_dir:$cwp_group:$cwp_prot:$cwp_tgp=
"CWP"@

Top Secret:1:$ts_dir:$ts_group:$ts_prot:$ts_tgp=
'TOPSECRET'@

Specat:1:$specat_dir:$specat_group:$specat_prot:$specat_tgp=
'SPECAT'@

Limdis:1:$limdis_dir:$limdis_group:$limdis_prot:$limdis_tgp=
'LIMDIS'@

AMHS Test:1:amhstest:amh_test:750:10=
"TESTAMHS";"TEST_AMHS";'aaa'@

Exclusive For:1:$excl_dir:$excl_group:$excl_prot:$excl_tgp=
'EXCLUSIVEFOR';"Exclusive";`WILLI*`@

Personal For:1:pers:amh_excl:750:11=
'PERSONALFOR'@

Fbis:1:$fbis_dir:$fbis_group:$fbis_prot:$fbis_tgp=
"FMFBIS"@

Nato:1:$nato_dir:$nato_group:$nato_prot:$nato_tgp=
"NATO"@

No Contractors:1:$nocon_dir:$nocon_group:$nocon_prot:$nocon_tgp=
'NOCONTRACT'@
```

2.3.5 The VARDEF File

The operation of the processes that send data from the SAT/CBT directory structure to the TOPIC database (**sat_feed**, **cbc_feed**), and deliver messages to the Message Manager client (**profile_hit**), are controlled by entries in the AMHS **vardef** (variable definition) file. The SAT/CBT and comeback copy (CBC) feed processes as well as the **profile_hit** executable have multiple modes of operation. These modes of operation have been developed to meet the needs of the various sites at which the AMHS has been deployed. The **vardef** file eliminates the need to deliver custom feeds and supporting profile distribution executables for each site. The AMHS Server processes read the **vardef** file to determine the mode of operation they should assume. The Message Manager client in the PC/NT version also accesses the **vardef** file to determine if it should default to an E-mail or database table mode of delivery, and determines the directory path to support files. Following is the normal **vardef** file location and contents.

/h/AMHS/Server/Topic/amhs_db/vardef

```
#
# VARDEF: Updated 06/13/94 S. Scandore
#
#       Define AMHS configuration variables.
#
site=GCCS                # Define AMHS site
Dba_perms=700            # Topic DBA permissions for log dirs.
Dba_group=gccs
```

```
max_wait=10              # Define maximum time feed will wait.
poll_frequency=30 # Feed poll time.

#
# The following variables are used to setup
# the section message processing parameters.
#
smpEnabled=TRUE

smpZwait=0
smpYwait=0
smpOwait=5
smpPwait=10
smpRwait=15

#
# Define move_token to specify the directory
# location where the processed message tokens
# are copied. They are copied after processing.
# If this is variable is not defined, the token
# will be deleted after processing.
#
#move_token=/usr/topic/amhs_db/second_sat
#
# Define add_dot to describe how dots are
# added to the message lines.
#
#    add_dot=all      // Adds dots to all pla lines + BT line.
#    add_dot=ftib     // Adds dots to the FROM, TO, INFO, BT lines.
#    add_dot=none // This is the default setting.
add_dot=ftib
#
# EXECUTABLES:
#
#           These define the lcoation of certain executables
#           used by the AMHS.
#
rt_bin=/h/COTS/Topic/current/bin
#
# CONVERT TOOLS:
#
#           These entries define the location of
#           dos to unix ocnversion tools.
#
dos_bin=/usr/ucb/dos
dos2unix=$dos_bin/dos2unix
unix2dos=$dos_bin/unix2dos
#
# SAT INTERFACE:
#
#           Define entries to support the SAT interface.
#
sat_dir=/h/AMHS/Server/sat
sat_exe_dir=/h/AMHS/Server/sat
rt_db=/h/AMHS/Server/topic/amhs_db
cfe_dir=/h/AMHS/Server/sat/autodin

error_dir=$rt_db/error
archive_dir=autodin/archive
sat_queue=$cfe_dir/archive
xmit_dir=$sat_dir/autodin/xmit
reject_dir=$sat_dir/autodin/reject
```

```
dac_lines=14
log_dir=$rt_db/log
dac_list=$rt_db/daclist

feed_dir=/h/AMHS/Server/dac
pc_path=T:\stop
#
# PUBLISH SUTFF:
#
#       Define the bahavior of the publish program.
#
topic_version=3.1.5;
weekly_pubs=WEDNESDAY,SUNDAY;
mailbox=$rt_db/mailbox;         # Topic mailbox directory
#
# FEED INFORMATION:
#
#       Define some SAT definitions.
#
sat_dp=dp1
cbc_dp=dp4
cbc_queue=$sat_queue;             # CBC feed directory

svr_cback=$rt_db/coord/comeback;      # cbc feed comeback directory
time_stamp=$sat_dir/cbc/time.dat;     # time stamp file for cbc feed
#
# TOPIC PART:
#
#       Define where the topic partitions are located.
#
out_partition_dir=$rt_db;         # dir containing dd173 partitions
adt_partition_dir=$rt_db;         # dir with adt partitions
gen_partition_dir=$rt_db;         # dir with genser autodin partitions
cbc_partition_dir=$rt_db;         # dir with cbc partitions
fbis_partition_dir=$rt_db;        # dir with fbis partitions

#
# DAC DIR:
#
#       Define where the dac directory is located.
#
dac_dir=/h/AMHS/Server/dac
dac=$dac_dir

gen_files_dir=$dac_dir/*/r;
cbc_files_dir=$dac_dir/*/r;
#
# DAC GROUP NAMES:
#
#       These definitions equate the AMHS groups name to
#       the corresponding UNIX group names.
#
cwp_group=amh_cwp
fbis_group=amh_fbis
excl_group=amh_excl
gen_group=gccs
limdis_group=amh_limd
nato_group=amh_nato
nocon_group=gccs
pers_group=amh_pers
specat_group=amh_spec
ts_group=amh_ts
special_grp=amh_spec
test_group=amh_spec
```

```
#
# DAC DIRECTORIES:
#
#         These definitions equate the AMHS group names to
#         the corresponding filesystem directory.
#
cwp_dir=cwp
fbis_dir=fbis
excl_dir=excl
gen_dir=general
limdis_dir=limdis
nato_dir=nato
nocon_dir=nocon
pers_dir=personal
specat_dir=specat
ts_dir=ts
special_dir=special
test_dir=special

#
# DAC PROTECTIONS:
#
#         These definitions define the file protections that
#         belong to each AMHS group.
#
cwp_prot=750
fbis_prot=750
excl_prot=750
gen_prot=750
limdis_prot=750
nato_prot=750
nocon_prot=750
pers_prot=750
specat_prot=750
special_prot=750
ts_prot=750
test_prot=750

#
# TOPIC GROUPS
#
#         These definitions define the topic group values.
#
all_tgp=16
gen_tgp=16

cwp_tgp=1
fbis_tgp=2
excl_tgp=3
limdis_tgp=4
nato_tgp=5
nocon_tgp=6
pers_tgp=7
specat_tgp=8
ts_tgp=9

#
# CAUTION WHEN EDITTING!!!!
#

my_pid=$$
host="amhserver"
year=`date '+%y'`
julian_day=`date '+%j'`
```

```
#
# GCCS TO PC AUTODIN message interface
#
# The email_notify flag is FALSE by default so
# that the profile_hit-gmail-table_deliv capability
# is DISABLED.
#
email_notify=FALSE
email_msg_type=whole_message
email_notify_prog=/h/AMHS/Server/progs/gmail
email_dir=/h/AMHS/Server/dac/emdir

dos_bat_mail_process=TRUE
email_envelope_dir=/h/AMHS/Server/dac/emenvdir
table_deliv_mount=/h/AMHS/Server/dac
```

2.3.6 AMHS Process Distribution

The distribution of AMHS software functions across multiple processes and CPUs has been designed to handle a wide variety of traffic loads by proper sizing of the AMHS Server without any software changes, while providing responsive file services to AMHS users on the LAN workstations. The AMHS process distribution is shown in Figure 2-20.

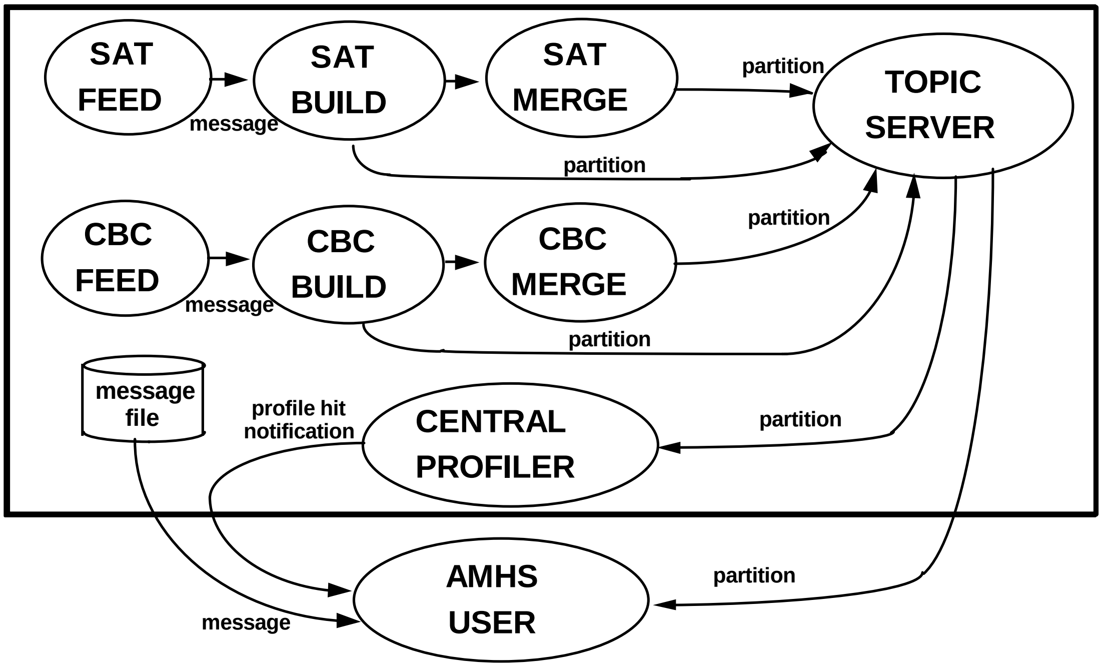


Figure 2-20. AMHS Process Distribution

The following paragraphs describe how the inbound message processing flow routes inbound message to the AMHS user's desktop.

2.3.7 SAT Feed Data Flow

AUTODIN messages are received into the AMHS by Cavalier Communications' SAT/CBT software. The messages are stored in the SAT/CBT archive directory structure and a notification token is placed in a backside queue directory polled by the **sat_feed** executable. The messages are then introduced into the Verity TOPIC Real-Time by the **sat_feed** as shown in Figure 2-21 and described below. As messages are received from the AUTODIN via the SAT/CBT they are stored in the ARCHIVE directory structure of the SAT/CBT under a directory corresponding to the Julian date. The SAT/CBT software places a token file in one of five directories corresponding to the precedence of the message. These backside queue (BSQ) directories are called **/h/AMHS/Server/sat/autodin/bsq1** through **/h/AMHS/Server/sat/autodin/bsq5**. The token file placed in these BSQs contain the path to the AUTODIN message just received. The **sat_feed** process on the AMHS SERVER node scans the BSQ directories in precedence order looking for these tokens, placed there by the SAT/CBT indicating that a new message has been received. When the **sat_feed** finds a token, it runs the message through a DAC filter (is site configurable in **daclist**) to determine if any special handling of the message is required, e.g. SPECAT or Personal For. The **sat_feed** also logs receipt of the message, FL2 and FL4 security information and any DAC applied to the message into a **sat_feed** audit log for that Julian date, e.g. sat_123.log. The SAT/CBT audit log is located in the AMHS Real-Time log directory **/h/AMHS/Server/topic/amhs_db/log**. The **sat_feed** puts a "period" (.) in front of lines beginning with "FROM, TO and INFO" until the first "BT" line of the message. Dotting parameters are set in the **vardef** file. Only the first occurrence of each word is marked. This is for the purpose of "zoning" a message, which is a technique used for profiling messages.

The **sat_feed** also creates a delivery record which is stored separate from the message file. The delivery record contains information such as the precedence, classification, time of receipt, DAC, and will contain a list of recipients once the message has been profiled by the TOPIC Central Profiler process. Finally, the message is ready for the TOPIC database. The **sat_feed** copies the processed message into appropriate DAC message directory, or creates the directory if it is not there, and tells TOPIC where the message is located and what DAC applies. The **sat_feed** uses two AMHS system tables to determine DAC for a message. These tables are **/h/AMHS/Server/topic/amhs_db/vardef** and **/h/AMHS/Server/topic/amhs_db/daclist**.

```
                            —>DAC DIR
AUTODIN—>sat_feed  —>dp1
messages         ↑

               daclist—Sets sat_feed parameters
               vardef—For daclist variable with leading `$’
```

As shown in Figure 2-19, the SAT places incoming messages in the J:\AUTODIN message files. The **sat_feed** takes the messages, preprocesses them and hands them to the DACDIR and dpt1 process. The **sat_feed** gets its parameters from the **daclist** files which in turn, in some cases, gets variable definitions from the **vardef** file.

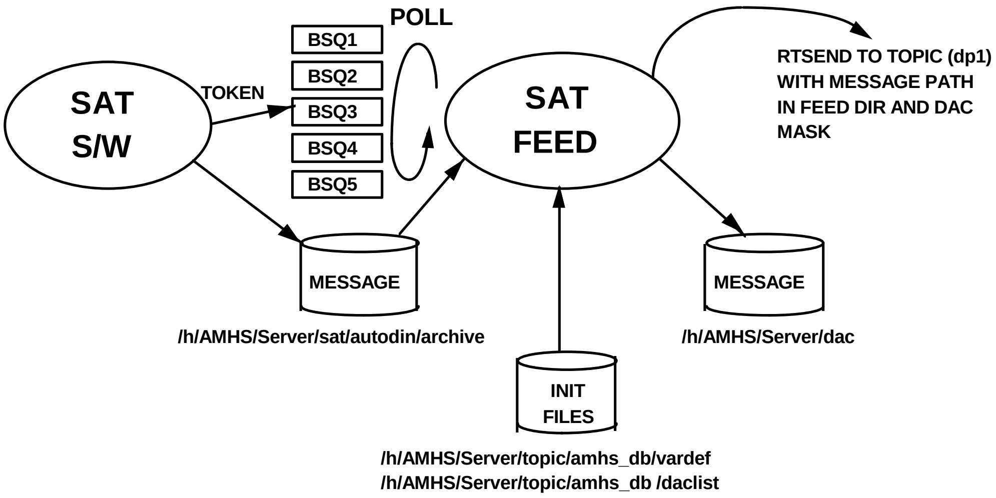

POLL
BSQ1
BSQ2
BSQ3
BSQ4
BSQ5
SAT S/W
TOKEN
SAT FEED
RTSEND TO TOPIC (dp1) WITH MESSAGE PATH IN FEED DIR AND DAC MASK
MESSAGE
/h/AMHS/Server/sat/autodin/archive
MESSAGE
/h/AMHS/Server/dac
INIT FILES
/h/AMHS/Server/topic/amhs_db/vardef
/h/AMHS/Server/topic/amhs_db /daclist

Figure 2-21. SAT Feed Message Flow

The roles of the **vardef** and **daclist** tables are described below.

(1) **vardef** - This table includes information about each DAC group pertaining to the location on disk where the message will be stored (group directory), the actual name of the DAC group and the bit mask used by the Verity TOPIC software to secure the message from being viewed by non-members of the DAC group. The **vardef** DAC group names and DAC directories define variables called out in the **daclist**. Paragraph 2.3.5 describes the **vardef** file in detail. For example:

```
# DAC GROUP NAMES:
#
#        These definitions equate the AMHS groups name to
#        the corresponding UNIX group names.
#
cwp_group=amh_cwp
fbis_group=amh_fbis
excl_group=amh_excl
#
# DAC DIRECTORIES:
#
#        These definitions equate the AMHS group names to
#        the corresponding filesystem directory.
#
cwp_dir=cwp
fbis_dir=fbis
excl_dir=excl
```

(2) **daclist** - This table includes the actual keywords which are searched for after the initial "break" (BT), Format Line 12 of the message. Each of these keywords is associated with a directory, group and mask defined in the **vardef** table. Only entries with a $ are referenced in the **vardef** file. For example:

```
Cwp:1:$cwp_dir:$cwp_group:$cwp_prot:$cwp_tgp=
"CWP"@

Top Secret:1:$ts_dir:$ts_group:$ts_prot:$ts_tgp=
'TOPSECRET'@

Specat:1:$specat_dir:$specat_group:$specat_prot:$specat_tgp=
'SPECAT'@

Limdis:1:$limdis_dir:$limdis_group:$limdis_prot:$limdis_tgp=
'LIMDIS'@

AMHS Test:1:amhstest:amh_test:750:10=
"TESTAMHS";"TEST_AMHS";'aaa'@

Exclusive For:1:$excl_dir:$excl_group:$excl_prot:$excl_tgp=
'EXCLUSIVEFOR';"Exclusive";`WILLI*`@

Personal For:1:pers:amh_excl:750:11=
'PERSONALFOR'@

Fbis:1:$fbis_dir:$fbis_group:$fbis_prot:$fbis_tgp=
"FMFBIS"@

Nato:1:$nato_dir:$nato_group:$nato_prot:$nato_tgp=
"NATO"@

No Contractors:1:$nocon_dir:$nocon_group:$nocon_prot:$nocon_tgp=
'NOCONTRACT'@
```

After determining if any DAC is required for the message, the **sat_feed** copies the message from the SAT/CBT archive directory structure (**/h/AMHS/Server/sat/autodin/archive/...**) to the feed directory structure (**/h/AMHS/Server/dac/....**) with the corresponding DAC directory and assigns the proper UNIX group. This is where the message is fed into TOPIC and where it remains for subsequent retrieval by the AMHS users. All the DAC directories defined in the **daclist** table are located under the FEED directory **/h/AMHS/Server/dac** (e.g. general, TS, SPECAT, etc.). In turn, under each of the DAC directories there are Julian date subdirectories corresponding with the SAT/CBT Julian date archive subdirectory (e.g., **/h/AMHS/Server/dac/general/r354/<*filename*>**). Once the message finds its resting place in the DAC directory structure, it is fed into TOPIC using the "make secure" mask from the DAC process. This mask will allow only those TOPIC users who are members of the TOPIC and UNIX group to which the message is being made secure to view the message either in a results browser or document ticker.

The TOPIC group assignment is determined by a user's entry in the **password** file, from where all TOPIC user privileges are determined. This file is edited and maintained with the Account Manage tool. For example:

```
user:   test01 ""
  /RTS = client
  /groups = 0,2,4,5
user:  test02 ""
  /RTS = client
  /groups = 0,1,2,3,4,5
```

The **sat_feed** feeds the message into TOPIC using the Verity-provided utility rtsend. The syntax of the rtsend call for a Top Secret (TS) message is as follows:

```
rtsend /h/AMHS/Server/topic/amhs_db/mailbox dp1 NEWTEXT /h/AMHS/Server/dac/ts/
r320 /<filename>   ts_mask
```

The rtsend call places a message in the mailbox of the SAT/CBT build (**dp1**) process to signal the build process to take the message in the **/h/AMHS/Server/dac/ ts/ r320 / <filename>** path and to introduce that message into TOPIC with the ts_mask as the TOPIC make secure mask. The TOPIC Real-Time mailbox directory structure is located in the **/h/AMHS/Server/topic/amhs_db** directory. The **/h/AMHS/Server/topic/amhs_db** directory is central to all TOPIC operations of the AMHS. For a more thorough discussion of this directory, see Chapter 3 (TOPIC Real-Time Directory Structure) in the TOPIC Real-Time Guide of the TOPIC Database Administrator's Guide V3.1.

2.3.7.1 Section Message Processing (SMP)

Section message processing in GCCS is performed in two different ways. The first is processing of the messages as they are received from AUTODIN. This is done by the **sat_feed** executable. The second method of processing is through a launch menu option within TOPIC. This is done from within the TOPIC database and is based upon the messages received within TOPIC.

When messages get too large, they are split into pieces (sections) by the AUTODIN Switching Center before transmission to the addressees. The **sat_feed** processing of the inbound message stream begins with the **sat_feed** detecting a sectioned message. The **sat_feed** holds the sectioned message in memory and either adds it to the other already-received sections or creates a new entry within memory. When all sections are received, the **sat_feed** consolidates the message and sends it on for processing by TOPIC. The sectioned message parts are then removed from memory. If SMP has been enabled via the **vardef** variable, the **sat_feed** will receive and store these sections one at a time until all sections can again be combined into one file for the user.

```
smpEnabled=TRUE;
```

A message is defined as being a “sectioned” message if either one of the following two strings is found in Format Line 12 of the message, **SECTION ss OF tt** or **FINAL SECTION OF tt**, where “ss” is the number of the current section and “tt” is the total number of sections for the entire message. If this information is detected, the **sat_feed** updates the SMP database kept in **/h/AMHS/Server/dac/SmpDB.dat**.

The SMP database is a file that can contain information about any number of outstanding sectioned messages. It contains a header record that describes general information about the message and then data records for each section of the message. This is a binary file read only by the **sat_feed** process.

SMP Message Flow: When the **sat_feed** detects a sectioned message, it queries the SMP database to see if the “family” (two messages with the same total number of sections and same message precedence and DTG belong to the same family) of this section already exists. If it does not yet exist, it is added to the SMP database. If it already exists, the section is added to the database for this section’s family. When all sections have been received, they are consolidated into one file. **sat_feed** then takes the consolidated file and treats it as a normal message by copying it to the “feed” directory, then sending it to TOPIC. The entire message family is then removed from the SMP database.

SMP Message Status: The outstanding sectioned messages are displayed in the **sat_feed** log. Say, for example, that a message containing 13 sections exists and that sections 9, 10 and 13 still have not arrived. The following 2 lines would appear at the **sat_feed** log:

```
FAMILY {dtg} {osri} {class} {#sections} {precedence} {first arrival time}
   Rcvd 10 of 13.  Missing 9-10, 13
```

SMP Alarms: How long should the **sat_feed** wait for all sections of a message to arrive? This depends upon the “precedence” of the message and may be altered by the following **vardef** variables (the # symbol is used for comments):

```
smpYwait=0;         # Emergency    0 = No wait
smpZwait=0;         # Flash        0 = No wait
smpOwait=30;        # Immediate    30 = 30 minutes
smpPwait=120;       # Priority     120 = 2 hours
smpRwait=1440;      # Routine      1440 = 24 hours
```

All time values are given only in minutes. Legal values range from 0 to 31680 minutes (22 days).

When the alarm value for a sectioned message expires, the message is consolidated with whatever messages *have* arrived, and sends the message on with "missing section indicators". For example:

>>>>> SECTION 09 OF 13 MISSING <<<<<

These missing section indicators are displayed in two places: 1) at the beginning of the message, and 2) at the position of the missing section.

If all the sections do not arrive within the defined wait time, the sections are passed to TOPIC anyway and can be processed separately.

2.3.7.2 Finding Sections in TOPIC

Note that individual segments of messages are not routinely passed to TOPIC for processing (except Emergency and Flash messages). It is only when parts of the message are missing or the delay in arrival is longer than the time-out period that individual sections are passed to TOPIC.

Under the launch pull-down menu within the TOPIC client is an option called "show all sections". The user can highlight a message within the message browser and select the "show all sections" option. This will start a secondary window that displays all the sections of the message received within TOPIC. In the case of sections appearing across partitions (e.g. different days), pressing the "merge" button will display the sections from all partitions. This does not create a composite. It only brings together all the sections of the message that are within the TOPIC database together in a single display.

Note that in the GCCS design, the **sat_feed** will have created a composite and not permitted any sections to be individually processed by TOPIC. So under ideal conditions, highlighting a message and selecting the "show all sections" will show only the composite, not the individual sections. If the **sat_feed** did not receive all the sections in a timely manner, then it is possible that sections held by the **sat_feed** timed out and were passed on to TOPIC for processing. In such cases some sections or partial grouping of sections may appear as a result of the "show all sections" command.

The "show all sections" command calls a script within the **/h/AMHS/Client/progs** subdirectory. This script builds a topic query with the correct parameters and submits it back to TOPIC for processing. Using an Xterm to execute this script will not work. It requires command line inputs and sends its output to a TOPIC mailbox for processing by the TOPIC server.

2.3.8 SAT/CBT Build Data Flow

The SAT/CBT Build data flow is shown in Figure 2-22. The SAT/CBT build process (**dp1**) running on AMHS Server retrieves the rtsend message from the **sat_feed** and builds the word list and document pointers for the message in the **/h/AMHS/Server/dac** subdirectory specified in the rtsend message. These pointers are stored in the TOPIC Real-Time directory

(**/h/AMHS/Server/topic/amhs_db**) as a TOPIC Real-Time "partition" subdirectory typically called **_dp1XXX** (where **XXX** is an alpha one up sequence, e.g. aaa, aab, aac, etc.). When the SAT/CBT build process completes building a new Real-Time partition, it notifies the SAT/CBT merge (**mg1**) and the TOPIC Real-Time Server (server) processes of the new data. The **dp1** process creates a log file in the **/h/AMHS/Server/topic/amhs_db** log directory containing the partition name it uses for the messages, all of the field information extracted and other useful information. Figure 2-22 PARTITION DATA refers to the Document Dataset and Document Index VDBs.

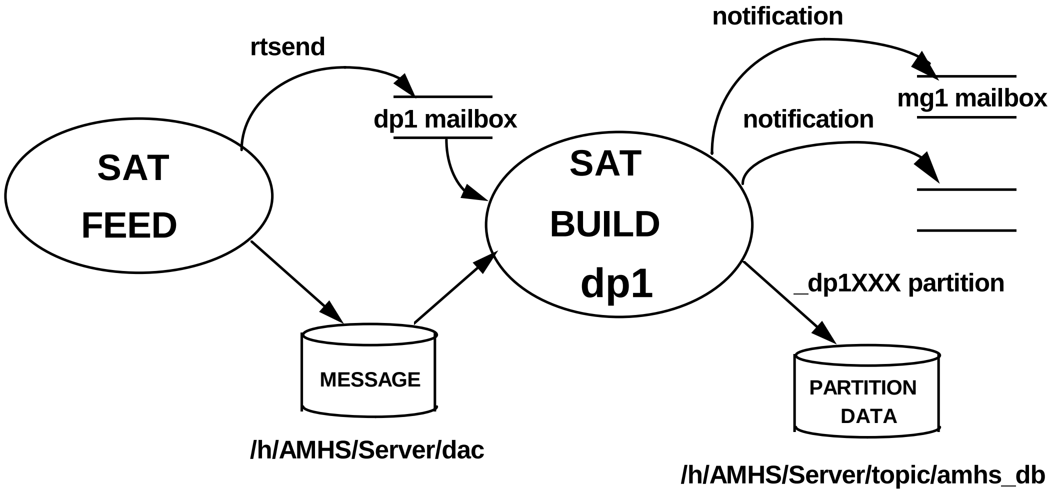


Figure 2-22. SAT/CBT Build Data Flow

2.3.9 SAT/CBT Merge Data Flow

The SAT/CBT Merger Data Flow is shown in Figure 2-23. The SAT/CBT merger process (**mg1**) receives messages from the SAT/CBT build process (**dp1**) when new real-time partitions (**_dp1XXX**) are available. The SAT/CBT Merger process consolidates the **_dp1XXX** partitions into larger, merged real-time partitions (**_mg1XXX**) and then notifies the TOPIC Real-Time Server (SERVER) of their availability. When a publish is done, all the real-time partitions are merged. The database automatically merges the real-time partitions based on the number of unpublished documents in the database. The AMHS publishes these partitions before they are automatically published in order to have control over the partition names used for archiving purposes and also for faster end user performance. The **mg1** process creates a log file in the **/h/AMHS/Server/topic/amhs_db** log directory containing useful information about publishes. Figure 2-23, PARTITION DATA refers to the Document Dataset and Document VDBs as well as real-time publishes.

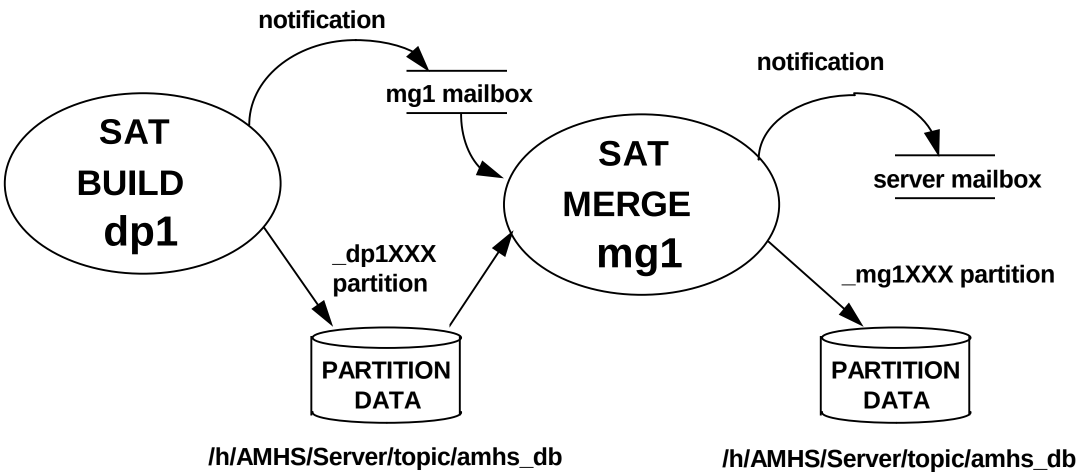


Figure 2-23. SAT/CBT Merger Data Flow

2.3.10 TOPIC Real-Time Server Data Flow

The TOPIC Real-Time Server (SERVER) is notified by the SAT/CBT build and merger processes whenever new partition data is made available and also in the case of the merger process what **_dp1** partitions being replaced by **_mg1** partitions. The SERVER process then broadcasts the availability of new partitions from the **dp1** process, and replacement partitions from the **mg1** process, to all TOPIC Real-Time users who are currently "logged on" to TOPIC. In this way, the SERVER process keeps the partition data used by all TOPIC users current. As new partition data are received from the SAT/CBT build process, and all TOPIC users are notified, the Central Profiler process of the TOPIC Real-Time System is also notified.

The TOPIC SERVER process validates the user login using the TOPIC password file (**password** gets compiled into **topic31.pwd**). The password file contains all the TOPIC processes like the TOPIC build process (**dp1**), the TOPIC Merger process (**mg1**) and the Topic Profiler (**pf0**,**pf2**,**pf3**...) as well as a list of every TOPIC user who has a database account. A TOPIC process must log on to the server before it can communicate any activities about the database. Every TOPIC user is also checked against this password file for TOPIC database access and privileges before any data is downloaded to the end user workstation. The SERVER process creates a log file in the **/h/AMHS/Server/topic/amhs_db/log** directory containing interprocess communication information. The message data flow of the TOPIC Real-Time SERVER is shown in Figure 2-24. The default profiler is **pf1**, though more profilers may be added if the system needs them. The profiler **pf0** profiles messages the API created to special queues.

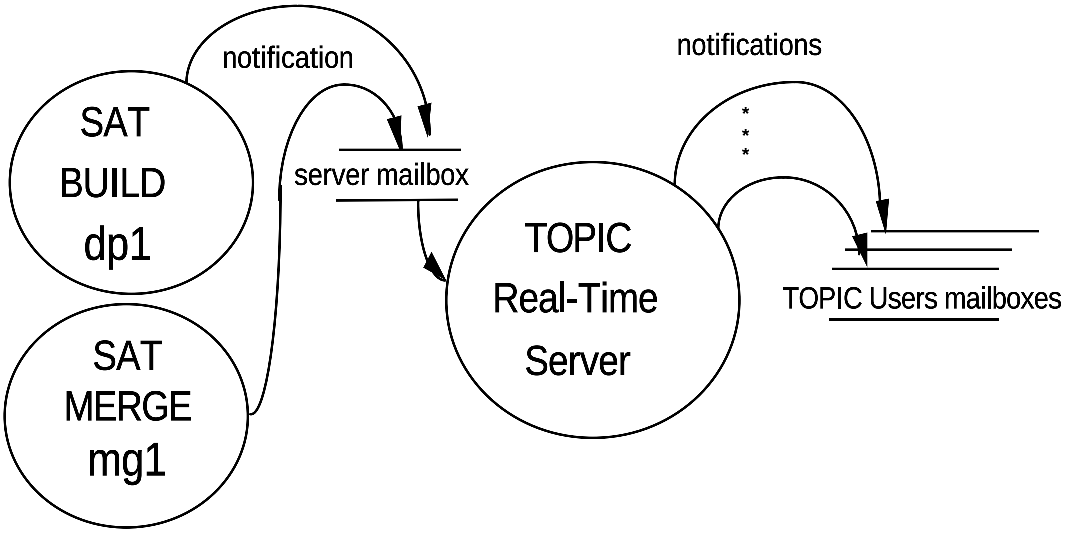


Figure 2-24. TOPIC Real-Time Server Inbound Message Data Flow

2.3.11 Central Profiler Data Flow

In Figure 2-25, the TOPIC SERVER is notified of new real-time partitions via the TOPIC SERVER mailbox directory. As partitions become available the TOPIC SERVER process notifies the Central Profiler process via its mailbox; there can be several profilers running, and for purposes of this discussion the Central Profiler will be referred to as "**pf1**". The **pf1** process performs an automatic retrieval function for AMHS users. It screens incoming messages according to selection criteria specified as queries, then notifies AMHS users as messages arrive which meet these selection criteria. AMHS users have ACTION and INFO TOPIC queues that are notified. When a "profile hit" occurs the delivery record created by the **sat_feed** for that message is updated with the user's name. The **pf1** process maintains a data file called **pf1.pfx** in the AMHS Real-Time directory which contains the last partition for the different feeds that was processed by it. The **pf1** process also creates a log file in the AMHS Real-Time log directory containing information on every partition screened as well as the user who was notified of a profile hit. In Figure 2-25, INIT FILES refers to the TOPIC configuration file **control.rts** and the **profiles** file (contains the list of users and their associated query).

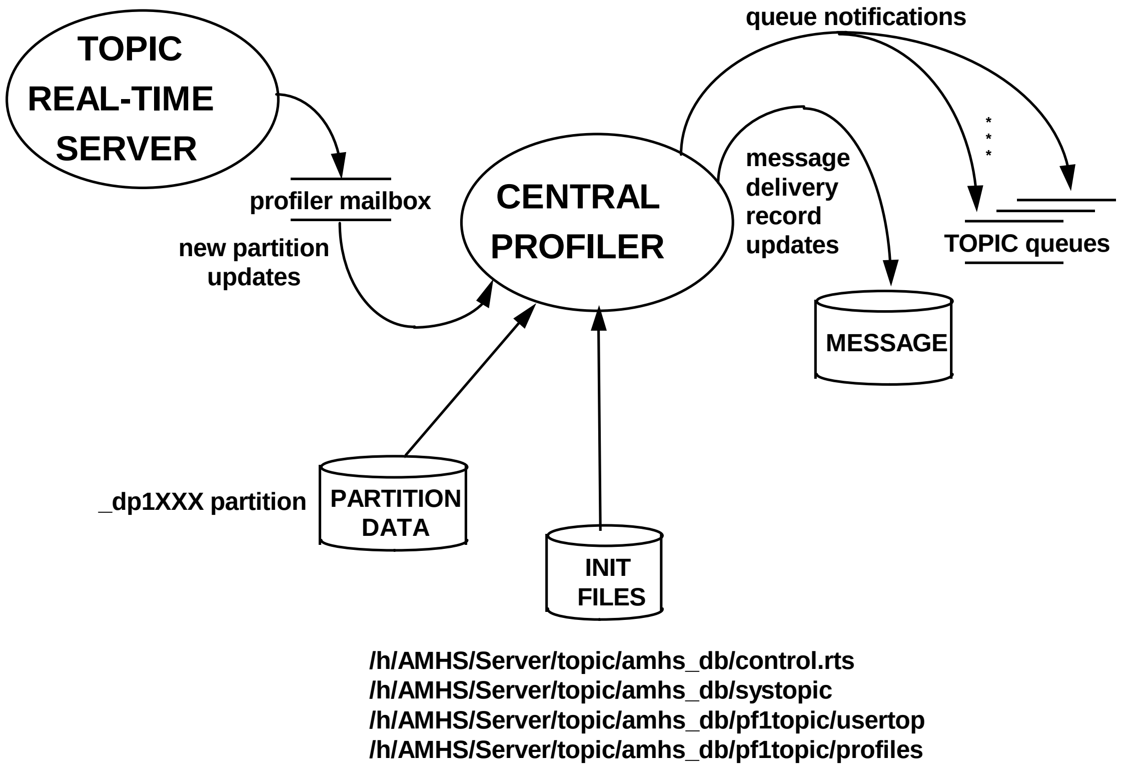

queue notifications
TOPIC REAL-TIME SERVER
profiler mailbox
new partition updates
CENTRAL PROFILER
message delivery record updates
* * *
TOPIC queues
MESSAGE
_dp1XXX partition
PARTITION DATA
INIT FILES
/h/AMHS/Server/topic/amhs_db/control.rts
/h/AMHS/Server/topic/amhs_db/systopic
/h/AMHS/Server/topic/amhs_db/pf1topic/usertop
/h/AMHS/Server/topic/amhs_db/pf1topic/profiles

Figure 2-25. Central Profiler Data Flow

2.3.12 TOPIC Client Message Retrieval Data Flow

Every process, including AMHS users, must log in to the TOPIC Server and be authenticated as a legitimate TOPIC Real-Time process. In Figure 2-26, users as well as the TOPIC processes are validated against the encrypted TOPIC password file. Users also have a unique mailbox, located under the TOPIC mailbox directory in **amhs_db**, through which they receive new partition and profile hit queue notifications. When a user logs in, the TOPIC Server process reads their preferences file in their TOPIC home directory which identifies who they are. After successful account validation, a list of all available partitions is read and a query is executed against the database, which retrieves all messages into the user's MESSAGE BROWSER. When the MESSAGE BROWSER is opened the message summaries are displayed in a Query Manager window in last-in-first-out order. On the other hand, the TOPIC message queue entries are not queries but rather directories which contain pointers to messages in the database which have been profiled to that specific user queue. AMHS users have three TOPIC message queues:

- ACTION - messages profiled by the Central Profiler (**pf1**) for action.
- INFO - messages profiled by the Central Profiler (**pf1**) for information only.
- MSGS_SENT - messages profiled by the Central Profiler (**pf1**) as comeback copies from successfully transmitted messages to AUTODIN.

The message retrieval client requests messages for display from the server, thereby eliminating the duplicate copies of messages across the network. The TOPIC Client data flow is shown in Figure 2-26. INIT FILES refer to the **topic.prf**, **master.prf** and **launch.lnc** files.

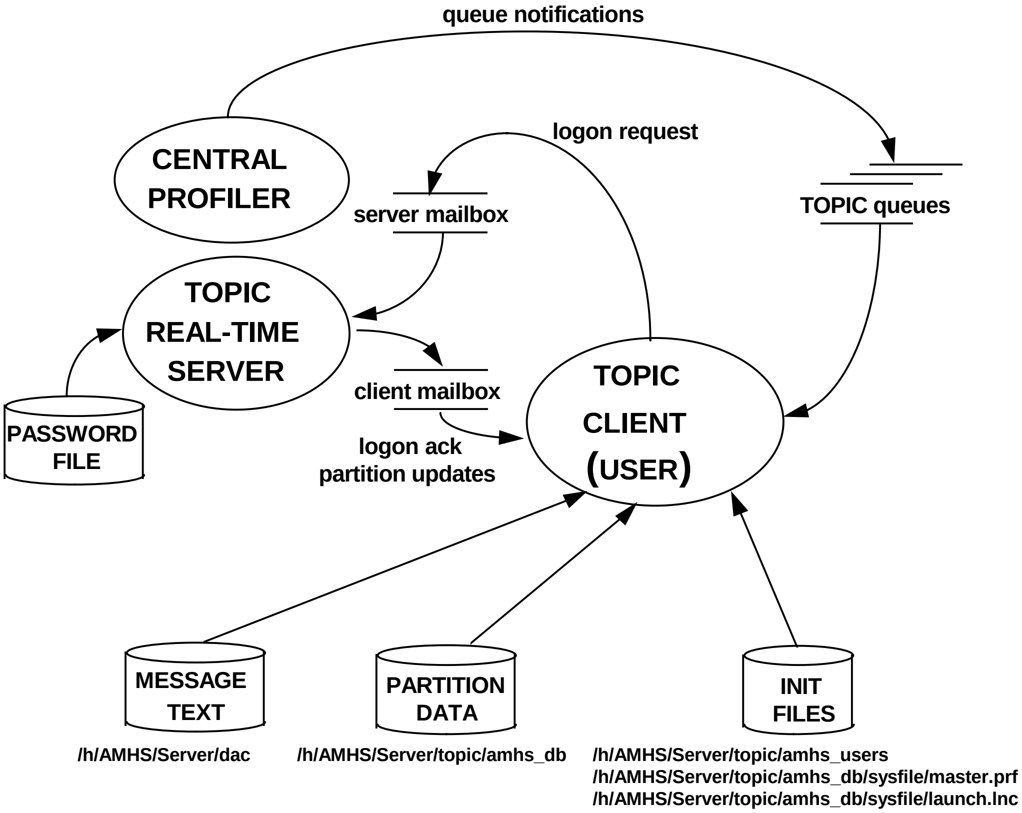


Figure 2-26. TOPIC Client Data Flow

2.3.13 Publishing Real-Time Partitions

Normal real-time operations of the TOPIC Partition Merger processes for the **sat_feed** (**mg1**) and **cbc_feed** (**mg4**), will take care of consolidating real-time partitions. However, for performance and archive reasons, topic real-time partitions need to be periodically combined into static partitions. This process is called “publishing” partitions. It is wise to convert all of the partitions generated daily into a single, published partition. The AMHS Server is configured to automatically publish real-time partitions once a day. The server’s automatic publishing feature is executed via a crontab entry on the AMHS Server. This crontab entry looks like the following:

```
45 23 * * * /h/AMHS/Server/Scripts/admin/DailyPublish > /dev/console
```

This entry instructs the AMHS Server to execute the **Dailypublish** script at one minute past midnight everyday of the week. Refer to the Solaris documentation for further information on crontab entries.

NOTE: The crontab entry may be modified if the configured publish time of 00:01 is not suitable for your site. This is the only site-configurable parameter for publishes.

Even though the publish is handled automatically, there are some important aspects about the publish that should be understood, if it becomes necessary to manually manipulate the TOPIC processes. First, all published partitions will have the following naming convention:

For incoming message partitions:	SAT{jday}V{sequence}
For comeback message partitions:	CBC{jday}V{sequence}

The {jday} is the current Julian day and the {sequence} represents a sequence number with “1” being the first publish of the day. In most cases there will only be one publish performed per day and the sequence number will never be greater than one. In very high traffic sites (>5k/day) it may be necessary to publish more than once a day to maintain topic response time.

Example: (the naming convention is required by the AMHS Sys Admin Tools)

For incoming messages on day 345:	SAT345V1
For comeback messages on day 345:	CBC345V1

It is very important to maintain this naming convention. In other words, published partitions should not be renamed and publishes should only be performed using the **DailyPublish** script which guarantees the naming convention. The naming convention is important for the backup and restore tools that are being integrated into the existing AMHS Sys Admin Tool.

The DailyPublish script creates a backup copy of the current partlist before performing its automatic publish. The current partlist is maintained on the server and is called:

/h/AMHS/Server/topic/amhs_db/partlist

The backup partlist is called:

/h/AMHS/Server/topic/amhs_db/partlist.previous

The backup partlist is kept in case the automatic publish fails and corrupts the current partlist. The old partlist can be recovered by renaming the backup partlist and restarting the TOPIC Server processes. Each day, a System Administrator or System Operator should check the publish log for any errors. This log is kept on the server and has the following naming convention:

/h/AMHS/Server/topic/amhs_db/log/publish.{jday}

The AMHS publish is kicked-off by a UNIX crontab entry daily, at 00:01. In TOPIC a real-time partition can be distinguished from published partitions by a preceding underscore character (_), e.g. **_mg10aaa**. The AMHS publish software uses its feed source (**sat_feed** or **cbc_feed**) and the Julian date in naming a newly published partition. An example for a daily publish for the **sat_feed** would be: **SAT345V1**. The same naming convention is used for the **cbc_feed**. When TOPIC is told to publish, it merges all of the real-time partitions created since the last TOPIC publish notification into the specified published partition. When the Partition Database Builder **(dp1** or **dp4**) receives a publish message, it completes its current activities and then forwards the publish notice to the Partition Merger (**mg1**, **mg4**), which actually performs the publishing.

2.3.14 Deleting Published Partitions

Your site policy on how many days to save on-line message traffic will determine when to back up and delete published partitions. The message backup and restore Sys Admin Tool is normally used to remove partitions. If there are any candidates, the TOPIC command rtsend with a *DELPART* class are used to remove partitions from the active partition list file in the **/h/AMHS/Server/topic/amhs_db** Real-Time directory called **partlist**.

2.3.14.1 GHOST PARTITIONS

Sometimes, real-time partitions do not complete all steps in the TOPIC Partition Merger (**mg1**, **mg4**) process and leave what is referred to as “ghost partitions”. They are identifiable in the **mg1.adt** audit log by greping for “ERROR” as well as by scanning the real-time directory and noticing that a partition from an earlier date did not get published properly. There are many theories on why this happens, but typically a “glitch” in network communications is the scapegoat. For those instances when the System Administrator identifies a “ghost partition”, the recovery has the following steps:

(1) Review the **mg1.adt/mg4.adt** audit logs to determine what happened to the partition and what is its status.

(2) Check the **partlist** file against the real-time and published partitions in the Real-Time directory.

(3) Manually merge the partitions.

NOTE: For manually merging published partitions please refer to the TOPIC Real-Time Administrator’s Guide V3.1, Chapter 9: TOPIC REAL-TIME Maintenance, 9-12b.

2.3.14.2 ADDING A SINGLE MESSAGE INTO THE DATABASE

The System Administrator will occasionally need to add a message to the TOPIC real-time message database. An example would be for processing a message that the **sat_feed** flagged as an error. Another example might be that your user community is requesting a message that has already been deleted. There are two different procedures for reintroducing a message back into the TOPIC database.

The first procedure is used to refeed the message into the TOPIC database. This procedure reenacts the sequence of steps that are performed when a new message is received at the SAT terminal. This is the simplest mechanism since it will rely on the feed process to guarantee the classification of the message.

This procedure can not be used if the **sat_feed** process flagged the message as an error. Using this procedure will simply cause the error to reoccur. Also, messages in the SAT archive have a SAT preamble on the first line of the file. The SAT preamble must exist for this refeed procedure to work. Execute the following steps to refeed a single AUTODIN message into the database:

(1) Ensure the message exists in the SAT archive. In other words, make sure the message exists in one of the **/h/AMHS/Server/sat/autodin/archive** subdirectories. If not, recover the message from tape or copy it from some other backup.

(2) Create a SAT message token in the backside queue. Assume that we have a message named **102345.001** in the **r245** sat archive subdirectory. Execute the following command to create the message token.

```
cd   /h/AMHS/Server/Scripts/admin/

CreateToken   r245/102345.001
```

This will create a message token in the backside queue of the SAT. The token is placed in the **/h/AMHS/Server/sat/autodin/bsq3** subdirectory. Once the token is created, the **sat_feed** process will feed the message into the TOPIC database. Check the current **sat_feed** log to ensure the message was handled properly.

The second procedure for reintroducing a message into the TOPIC database involves directly interacting with the TOPIC database server using the **rtsend** command. This procedure can be used to reintroduce a message that the **sat_feed** process could not handle.

This procedure sends messages that do not have the SAT preamble attached. Execute the following steps to introduce a message.

(1) Place the message file in a subdirectory within the **/h/AMHS/Server/dac** directory. You may wish to create a new directory for this type of situation. For example, you may want to create a directory called **/h/AMHS/Server/dac/manual** to place messages that are manually fed.

(2) Execute the rtsend command as the **amhs_dba**. Assume that we have a message file with the following filename: **/h/AMHS/Server/dac/manual/my_message**

```
cd  /h/AMHS/Server/topic/amhs_db
```

```
rtsend mailbox dp1 NEWTEXT /h/AMHS/Server/dac/manual/my_message  [mask]
```

The mask must be calculated. This is the one disadvantage to using this procedure. The **sat_feed** process automatically calculated the mask in the first procedure. There is a unique mask for each DAC defined group (e.g. LIMDIS, CWP, etc.). Calculate the mask by reviewing previous **sat_feed** log files. For example, assume our example was a SPECAT type message. Careful review of a **sat_feed** log would show the following information:

```
Access type:   Specat
TOPIC Mask:    0x00010000
```

This indicates that the mask for SPECAT messages is 0x00010000. This is a hex value and must be converted to a decimal number for the rtsend command. Given this example, the complete rtsend command would look like the following:

```
rtsend mailbox dp1 NEWTEXT /h/AMHS/Server/dac/manual/my_message 65536
```

Once completed, the message will appear in the TOPIC database.

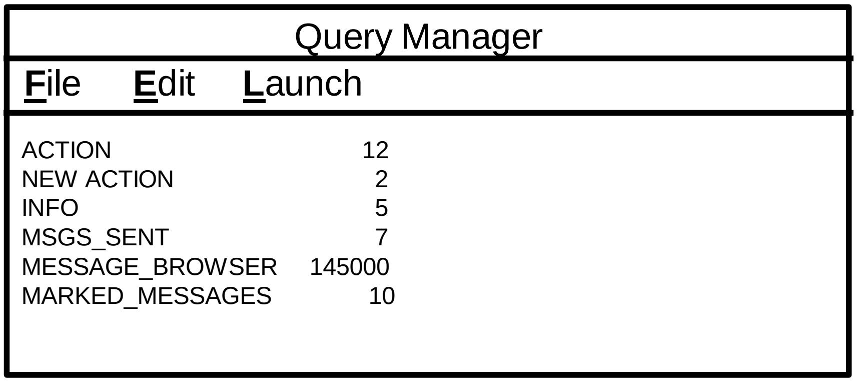
2.3.15 TOPIC Client Windows

The GCCS AMHS uses the commercial off-the-shelf (COTS) Verity Inc. TOPIC Real-Time product as its database engine. TOPIC Real-Time provides delivery of incoming AUTODIN messages to GCCS AMHS users' message queues as well as comeback copies of as-transmitted messages. As AUTODIN messages are received they are first put into the message database based on discretionary access control (DAC) filters and then profiled, using TOPIC queries, to one of several GCCS AMHS users' message queues. The TOPIC query is made up of conditions and keywords that match the user's "areas of interest" (AOIs). TOPIC queries are contained in a System Administrator controlled "profiles" file that is checked as each message is received. Upon message queue delivery, GCCS AMHS users can view the message queues in the TOPIC Query Manager Window, the TOPIC Message Browser Window to view a directory listing of those messages, and finally the Message Browser:AUTODIN Window to view the message itself. TOPIC Real-Time users may retrospectively search the entire message database using simple queries with "AND", "OR", or "NOT" conditions.

The layout of the TOPIC Windows is controlled by entries in two data files located in the **/h/AMHS/Server/topic/amhs_db/sysfile** directory; **master.prf** and **launch.lnc**. To launch the TOPIC Client, the GCCS AMHS users' local preference file, **topic.prf**, is used, which contains the user's TOPIC ID and points to the **master.prf** and **launch.lnc** data files.

2.3.15.1 TOPIC Query Manager Window

Upon successful activation, a small window is displayed for about 5 seconds containing TOPIC copyright information, version of software and user authentication. Message partition information is actually being downloaded as the TOPIC Query Manager Window opens as shown in Figure 2-27. The TOPIC Query Manager Window is the "main" window of the TOPIC Client application displaying the various message queues and the number of messages in each queue as well as the total number of messages in the database. The ACTION queue will contain messages profiled for a user to take action on. The NEW_ACTION queue will contain messages that have been forwarded, for action or information, by another user. The INFO queue will contain messages that were profiled for information only. The MSGS_SENT queue typically contains the "comeback copies" of as-transmitted messages to AUTODIN. Comeback copies typically will be sent to the Message Drafter and Message Coordination Points as well as the Message Releaser. The MESSAGE_BROWSER contains all messages that the particular user has discretionary access to; DAC is accomplished with UNIX group permission as well as TOPIC group permissions. The MARKED_MESSAGES queue is a storage area for messages identified from other queues which a user has selected to save off in this queue. To view the contents of any message queue simply highlight the message queue and press <ENTER> on the keyboard, or double-click the queue name.



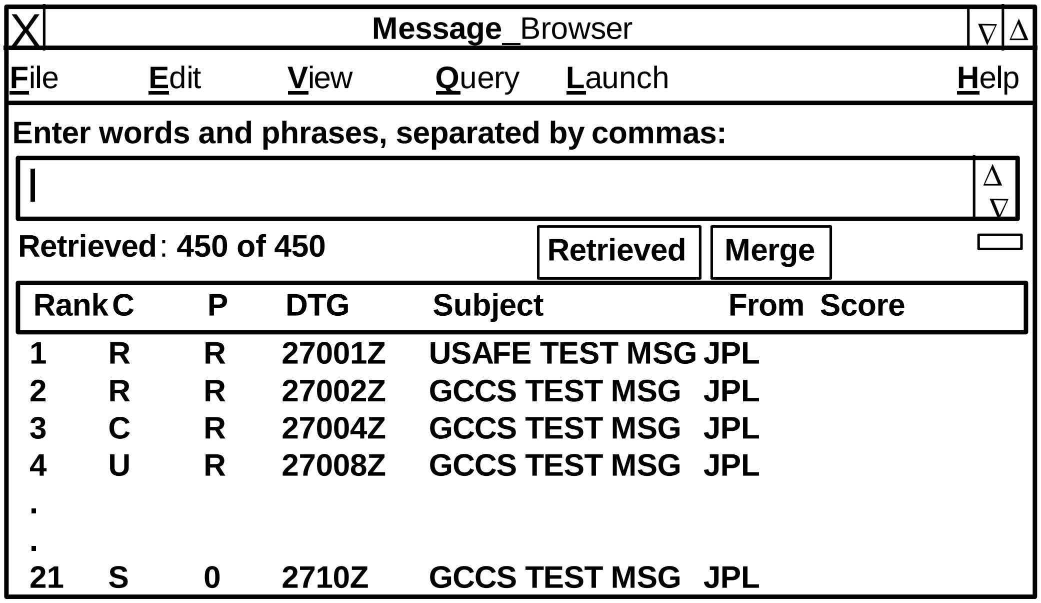
ACTION	12
NEW ACTION	2
INFO	5
MSGS_SENT	7
MESSAGE_BROWSER	145000
MARKED_MESSAGES	10

Figure 2-27. TOPIC Query Manager Window

For further information on TOPIC Query Manager Window and menu features please refer to the Verity TOPIC Users' Guide for Motif V3.1.

2.3.15.2 TOPIC Message Browser Window

The TOPIC Message Browser Window will display a summary listing of the messages for the queue that was selected. The columns of information for each message are defined in the 'styles' directory for the sat_feed process and the cbc_feed process. This directory of information is located in **/h/AMHS/Server/topic/amhs_db/styles/jsty** or **csty** ("**jsty**" stands for JANAP 128 style and "**csty**" stands for comeback copy style). The styles directory files contain the field information and parsing rules used when the message partitions are being built by the **dp1** and **dp4** TOPIC processes. These must not be manually edited because the TOPIC database entries will be corrupted. When it comes time to look at messages in the Message Browser Window the **master.prf** file and **launch.lnc** data files reference these fields of information. In Figure 2-21 the column titles mean: "C" for CLASSIFICATION, "P" for PRECEDENCE, "DTG" for date-time-group , "Subject" for subject line, and "From" for whom the message is from. The "Score" is an optional TOPIC feature which is used to "rank" messages in sequential order. AMHS sites should not use the ranking feature for AUTODIN message routing to user queue. If needed, a user could build special TOPIC queries to search for specific information. To view the contents of any message simply highlight the message and press <ENTER> on the keyboard or double-click the message.



Enter words and phrases, separated by commas:

Retrieved: 450 of 450

Rank	C	P	DTG	Subject	From	Score
1	R	R	27001Z	USAFE TEST MSG	JPL	
2	R	R	27002Z	GCCS TEST MSG	JPL	
3	C	R	27004Z	GCCS TEST MSG	JPL	
4	U	R	27008Z	GCCS TEST MSG	JPL	
.						
.						
21	S	0	2710Z	GCCS TEST MSG	JPL	

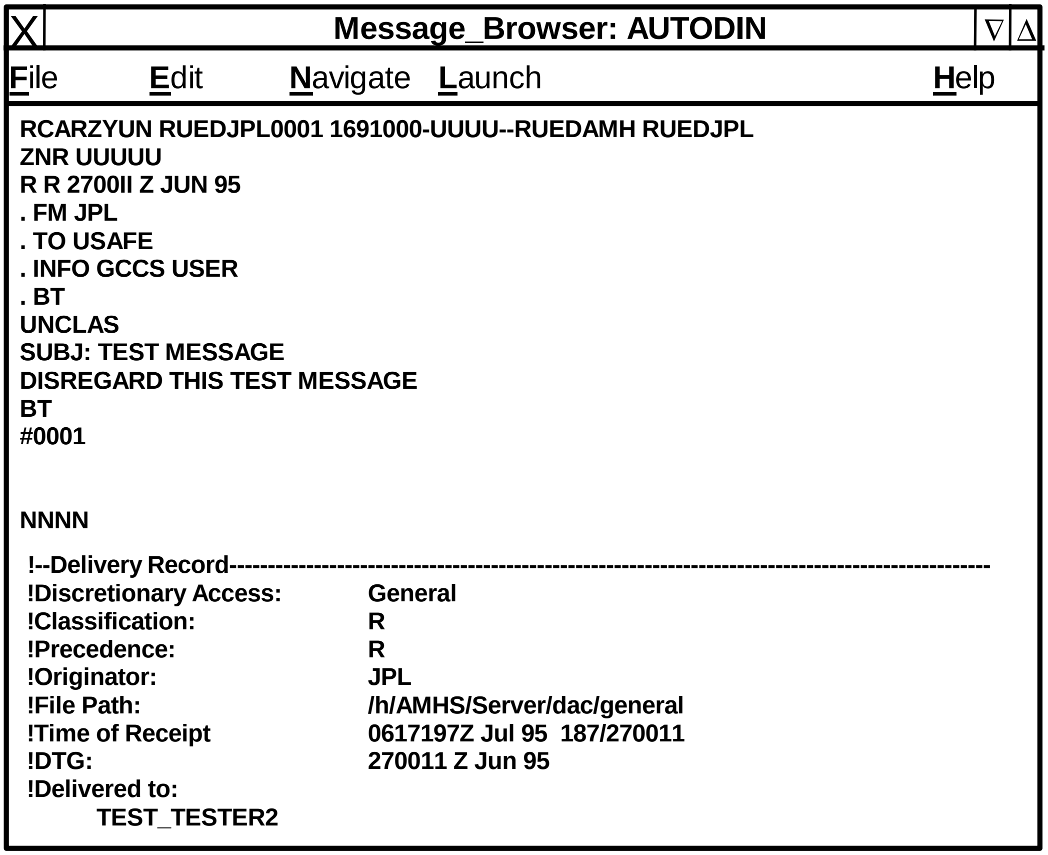
- SELECT THE MESSAGE BROWSER (CONTAINS ALL MESSAGES IN THE THE DATABASE TO DISPLAY A DIRECTORY LISTING OF MESSAGES.
- SELECT A MESSAGE TO DISPLAY BY DOUBLE CLICKING THE MOUSE.

Figure 2-28. TOPIC Message Browser Window

The TOPIC Message Browser Window, Figure 2-28, has an input area for "simple queries" into the message database. A user can enter keywords with optional "AND", "OR", or "NOT" condition. For further information on how to use the TOPIC Message Browser Window and menu features please refer to the Verity TOPIC Users' Guide for Motif V3.1.

2.3.15.3 TOPIC Message Browser: AUTODIN Window

The TOPIC Message Browser: AUTODIN Window, Figure 2-29, contains the AUTODIN message and its delivery record. These are actually two separate files but are displayed together. TOPIC does not allow modification to the message once it gains control of the message—when the **sat_feed** notifies the TOPIC Database Builder (**dp1**) process. The **sat_feed** optionally will implement “zoning” (through the **vardef** file) which is a period (.) at the beginning of the line starting at “FM” to the first instance of “BT”. This enables PLA profiling control and is the only modification made to any AUTODIN message in the GCCS AMHS. The delivery record for the message is also created during the sat_feed processing. The **sat_feed** completes the information up to “Delivered to” field which is reserved for the “profile process” routine. The “Delivered to” information is updated by the Central Profiler process when it calls the “**profile_hit**” routine. The parameters which are passed to **profile_hit** are: PREC, SOURCE, USER, DOC, DOC_FN, SUBJECT, and NOTIFICATION.

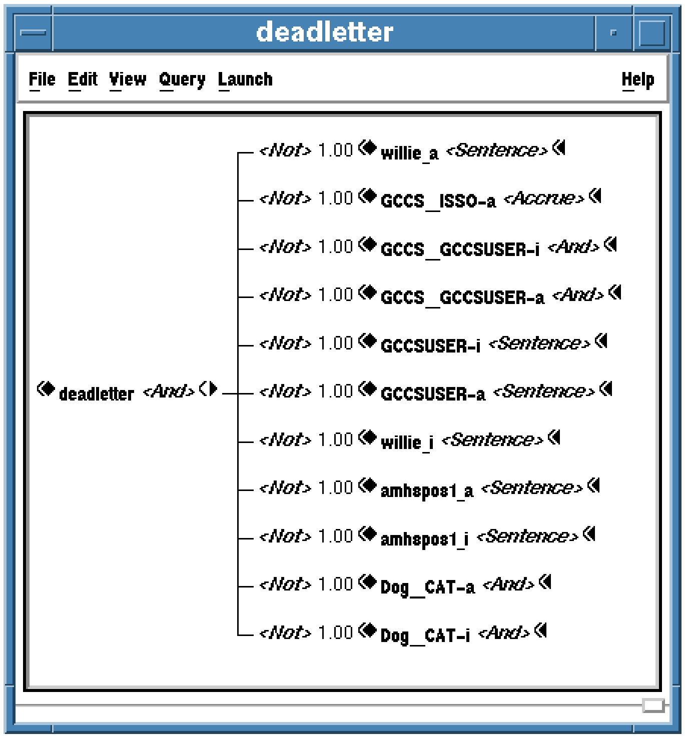


```
RCARZYUN RUEDJPL0001 1691000-UUUU--RUEDAMH RUEDJPL
ZNR UUUUU
R R 2700II Z JUN 95
. FM JPL
. TO USAFE
. INFO GCCS USER
. BT
UNCLAS
SUBJ: TEST MESSAGE
DISREGARD THIS TEST MESSAGE
BT
#0001


NNNN
 !--Delivery Record-----------------------------------------------------------------------------------------------
 !Discretionary Access:          General
 !Classification:                R
 !Precedence:                    R
 !Originator:                    JPL
 !File Path:                     /h/AMHS/Server/dac/general
 !Time of Receipt                0617197Z Jul 95  187/270011
 !DTG:                           270011 Z Jun 95
 !Delivered to:
         TEST_TESTER2
```

Figure 2-29. TOPIC Message Browser: AUTODIN Window

For information on how to use the menu features please refer to the Verity TOPIC Users’ Guide for Motif V3.1.

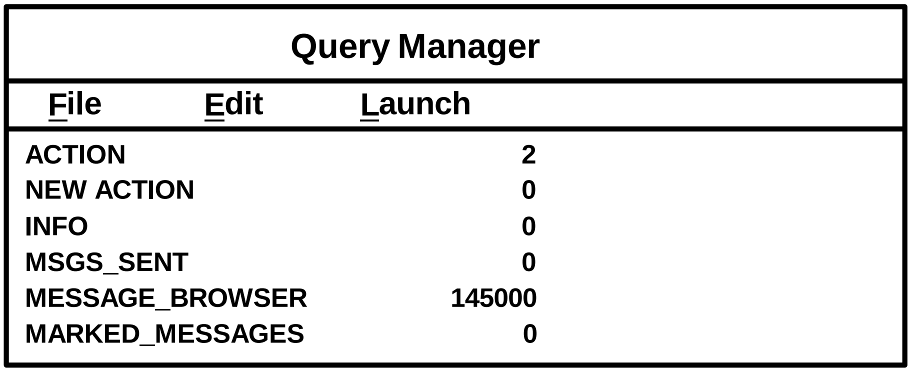
2.3.16 Dead Letter File



deadletter <And>
- <Not> 1.00 willie_a <Sentence>
- <Not> 1.00 GCCS_ISSO-a <Accrue>
- <Not> 1.00 GCCS_GCCSUSER-i <And>
- <Not> 1.00 GCCS_GCCSUSER-a <And>
- <Not> 1.00 GCCSUSER-i <Sentence>
- <Not> 1.00 GCCSUSER-a <Sentence>
- <Not> 1.00 willie_i <Sentence>
- <Not> 1.00 amhspos1_a <Sentence>
- <Not> 1.00 amhspos1_i <Sentence>
- <Not> 1.00 Dog_CAT-a <And>
- <Not> 1.00 Dog_CAT-i <And>

Each AUTODIN message received in the GCCS AMHS must be delivered to a human for disposition. Non-delivery of a message could occur if GCCS AMHS users' TOPIC queries were not robust enough to ensure a legitimate destination for a message. This circumstance would provide for a message just being dropped into the message database. To avoid this situation a "dead letter file" needs to be created. This is accomplished by creating a TOPIC query for a maintenance account (**amhs_oper** as an example) that contains a "NOT" condition on each user query in the profiles file used by the TOPIC Central Profiler(s). The NOT condition query would deliver any message not properly profiled to a GCCS AMHS user to the **amhs_oper** ACTION queue (or any one you choose). All of the users' INFO and ACTION queue profiles, topic trees that will profile messages to those queues, are the ones to include in the deadletter topic. MSGS_SENT and API interfaces need not be included. Some thought should go into the account that will be used to maintain the "dead letter file" because the account will need wide open UNIX privileges to access all message types as well as a PROJECT/POSITION pairing in order to run Message Manager to create a buckslip.

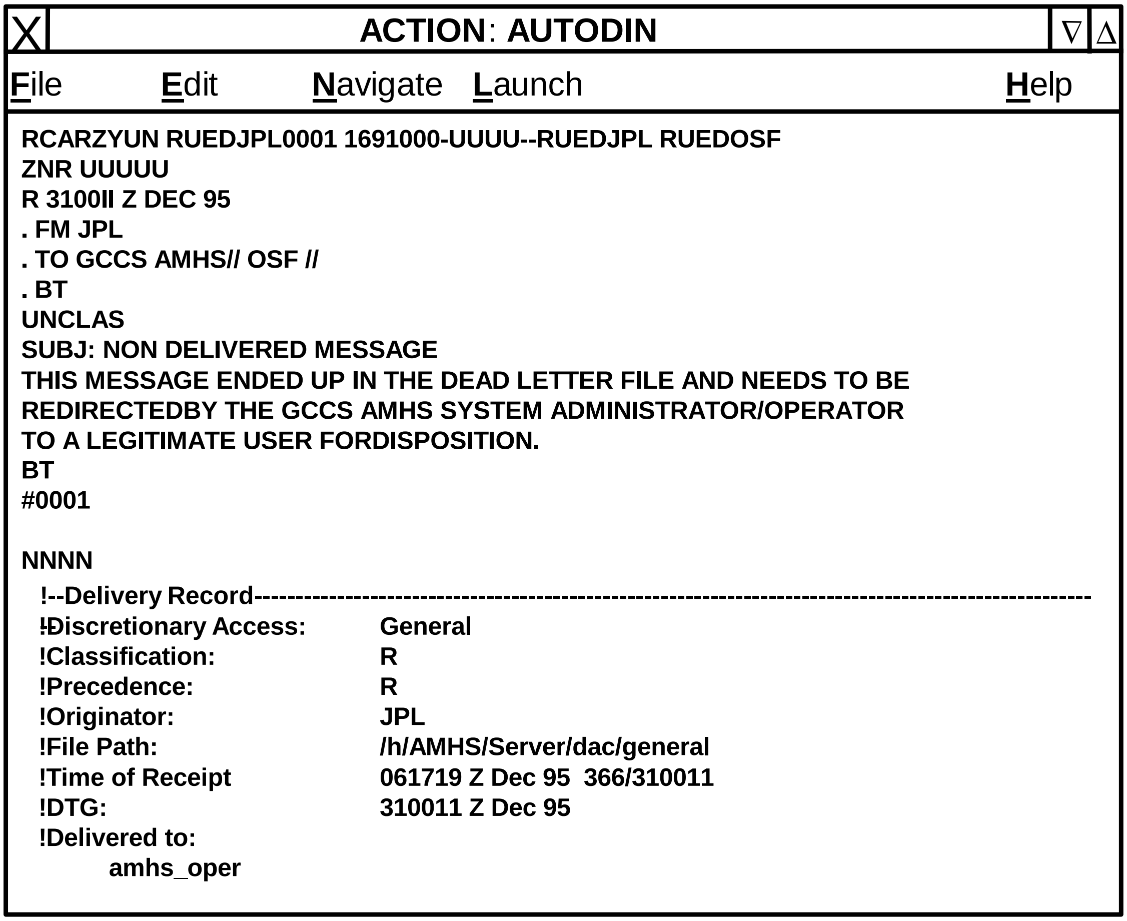
The **amhs_oper** account would log into TOPIC daily and check the ACTION queue for these messages. Figure 2-30 depicts two messages profiled to the **amhs_oper** account ACTION queue in the TOPIC Query Manager Window for further disposition.



ACTION	2
NEW ACTION	0
INFO	0
MSGS_SENT	0
MESSAGE_BROWSER	145000
MARKED_MESSAGES	0

Figure 2-30. TOPIC Query Manager Window

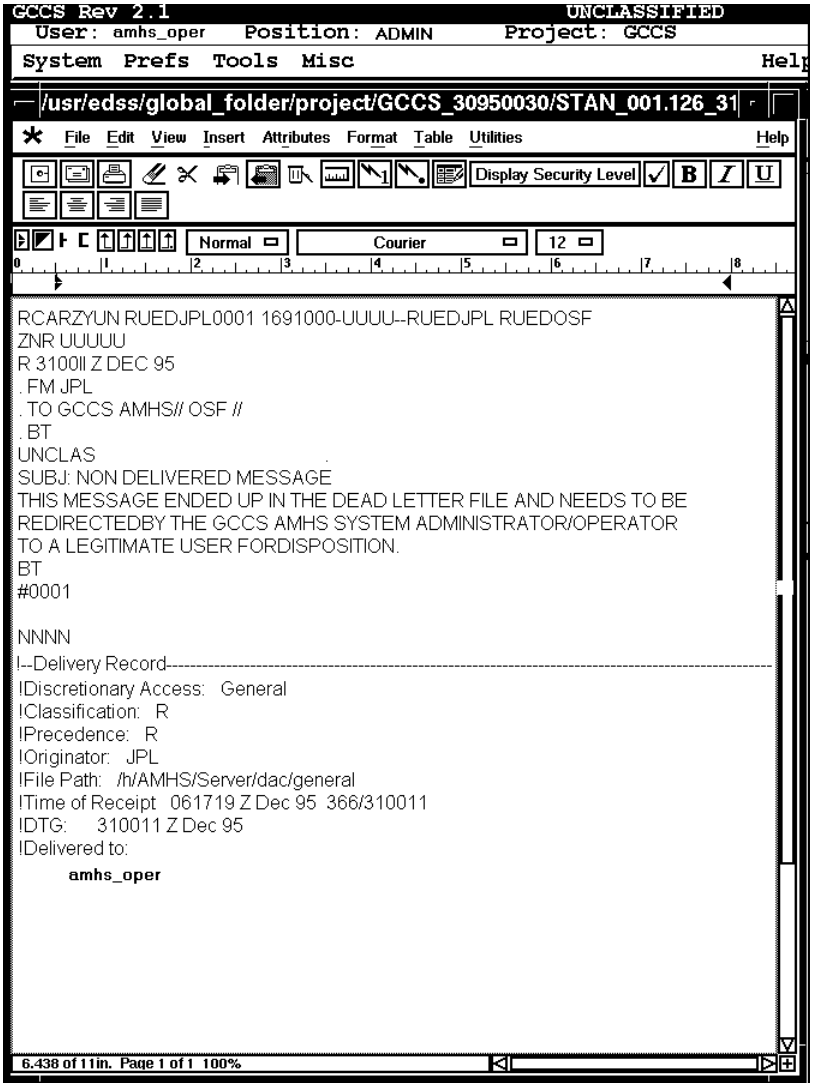
The **amhs_oper** would open the ACTION queue to view the messages. Figure 2-31 displays the message and delivery record in ACTION:AUTODIN Window. The delivery record indicates that the message was sent only to the **amhs_oper** account.



```
RCARZYUN RUEDJPL0001 1691000-UUUU--RUEDJPL RUEDOSF
ZNR UUUUU
R 3100II Z DEC 95
. FM JPL
. TO GCCS AMHS// OSF //
. BT
UNCLAS
SUBJ: NON DELIVERED MESSAGE
THIS MESSAGE ENDED UP IN THE DEAD LETTER FILE AND NEEDS TO BE
REDIRECTEDBY THE GCCS AMHS SYSTEM ADMINISTRATOR/OPERATOR
TO A LEGITIMATE USER FORDISPOSITION.
BT
#0001

NNNN
  !--Delivery Record------------------------------------------------------------
  !Discretionary Access:     General
  !Classification:           R
  !Precedence:               R
  !Originator:               JPL
  !File Path:                /h/AMHS/Server/dac/general
  !Time of Receipt           061719 Z Dec 95  366/310011
  !DTG:                      310011 Z Dec 95
  !Delivered to:
        amhs_oper
```

Figure 2-31. ACTION: AUTODIN Window

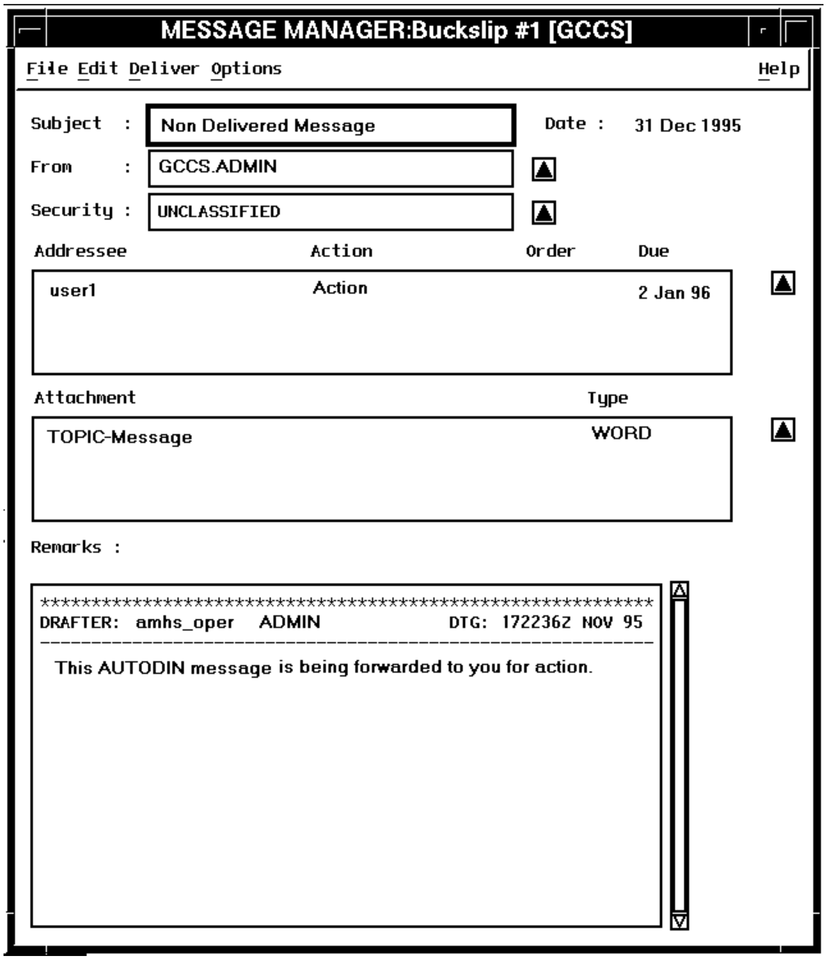
To route this message to a GCCS AMHS user the **amhs_oper** needs to create a buckslip in Message Manager and attach the message. The message should be copied into Applix Word and saved in a private folder first. The System Operator must know the user is authorized to see this message, because this method does not preserve DAC controls. Figure 2-32 shows the message after it has been cut and pasted from the ACTION:AUTODIN Window into the Applix Word Window.


GCCS Rev 2.1
UNCLASSIFIED
User: amhs_oper
Position: ADMIN
Project: GCCS
System Prefs Tools Misc
Help
/usr/edss/global_folder/project/GCCS_30950030/STAN_001.126_31
File Edit View Insert Attributes Format Table Utilities
Help
Display Security Level
B I U
Normal
Courier
12
6.438 of 11in. Page 1 of 1 100%

```
RCARZYUN RUEDJPL0001 1691000-UUUU--RUEDJPL RUEDOSF
ZNR UUUUU
R 3100II Z DEC 95
. FM JPL
. TO GCCS AMHS// OSF //
. BT
UNCLAS
SUBJ: NON DELIVERED MESSAGE
THIS MESSAGE ENDED UP IN THE DEAD LETTER FILE AND NEEDS TO BE
REDIRECTEDBY THE GCCS AMHS SYSTEM ADMINISTRATOR/OPERATOR
TO A LEGITIMATE USER FORDISPOSITION.
BT
#0001

NNNN
!--Delivery Record------------------------------------------------------------
!Discretionary Access:   General
!Classification:   R
!Precedence:   R
!Originator:   JPL
!File Path:   /h/AMHS/Server/dac/general
!Time of Receipt   061719 Z Dec 95  366/310011
!DTG:      310011 Z Dec 95
!Delivered to:
        amhs_oper
```

Figure 2-32. AUTODIN Message in Applix Words Window

The buckslip can be created now with directions in the “Remarks:” section that the attached message requires the GCCS AMHS user’s attention. Figure 2-33 shows the buckslip with possible instructions for the user.



Subject : Non Delivered Message Date : 31 Dec 1995

From : GCCS.ADMIN

Security : UNCLASSIFIED

Addressee	Action	Order	Due
user1	Action		2 Jan 96

Attachment	Type
TOPIC-Message	WORD

Remarks :

```
*****************************************************************
DRAFTER:  amhs_oper   ADMIN            DTG:  172236Z NOV 95
-----------------------------------------------------------------
  This AUTODIN message is being forwarded to you for action.
```

Figure 2-33. Message Manager Buckslip Window

If a message is unique, e.g. of probable one-time use, the operator will only need to buckslip the message. If the operator feels this is a message that should be profiled, they should include in the buckslip a note for the user to contact the AMHS System Administrator to include routing criteria in their user profiles.

With proper planning, the System Operator could use the reroute feature to forward messages to users. It should be remembered, however, that the user should have DAC privileges matching the message. Otherwise, the user will not be able to see or read the message. If you are a member of the topic group but not the UNIX group, the message *number* will appear in your queue but not the *message*.

2.4 TOPIC MESSAGE ROUTING CRITERIA

The GCCS AMHS is a knowledge based AUTODIN message delivery system. Messages have historically been authored and delivered from Communications Center to Communications Center with operators and runners at each end to assist the messaging process. Incoming messages are typically parsed and delivered to a point of contact in each organization who will again parse and deliver based on their knowledge of the organization. The first step to developing a plan, and subsequently a routing criteria as a collection of topic profiles, is to work with the using organizations and learn how messaging has been handled in the past. With your knowledge of the AMHS capability, you and your customers should be able to develop a working system quickly. It is essential for both accountability and reliability that you incorporate the dead letter account. This will also assist in tuning the profiling criteria to improve the delivery process.

2.4.1 Message Profiles

A **message profile** is a topic which has been built by the Administrator for the user. The central profiler performs standing queries with these topics against new messages. Message Profile building is an iterative process. The goal is to fully isolate the messages which are addressed to or are of interest to a particular user or office. The basic form of the profiles file is:

profile = '<FILTER> (DOCSOURCE = AUTODIN) AND GCCS_GCCSUSER -a'

GCCS_GCCSUSER -a' = the name of the topic.

The AMHS Administrator builds sets of message profiles specific to a site's operational requirements. Care must be taken not to leave any profiles without at least a default topic or you might route everything to that user. An installed AMHS is initially devoid of message profiles except for the two default profiles in the **/h/AMHS/Server/topic/amhs_db/pf1topic/profile**. The following listing, Figure 2-34 is a typical default profile file.

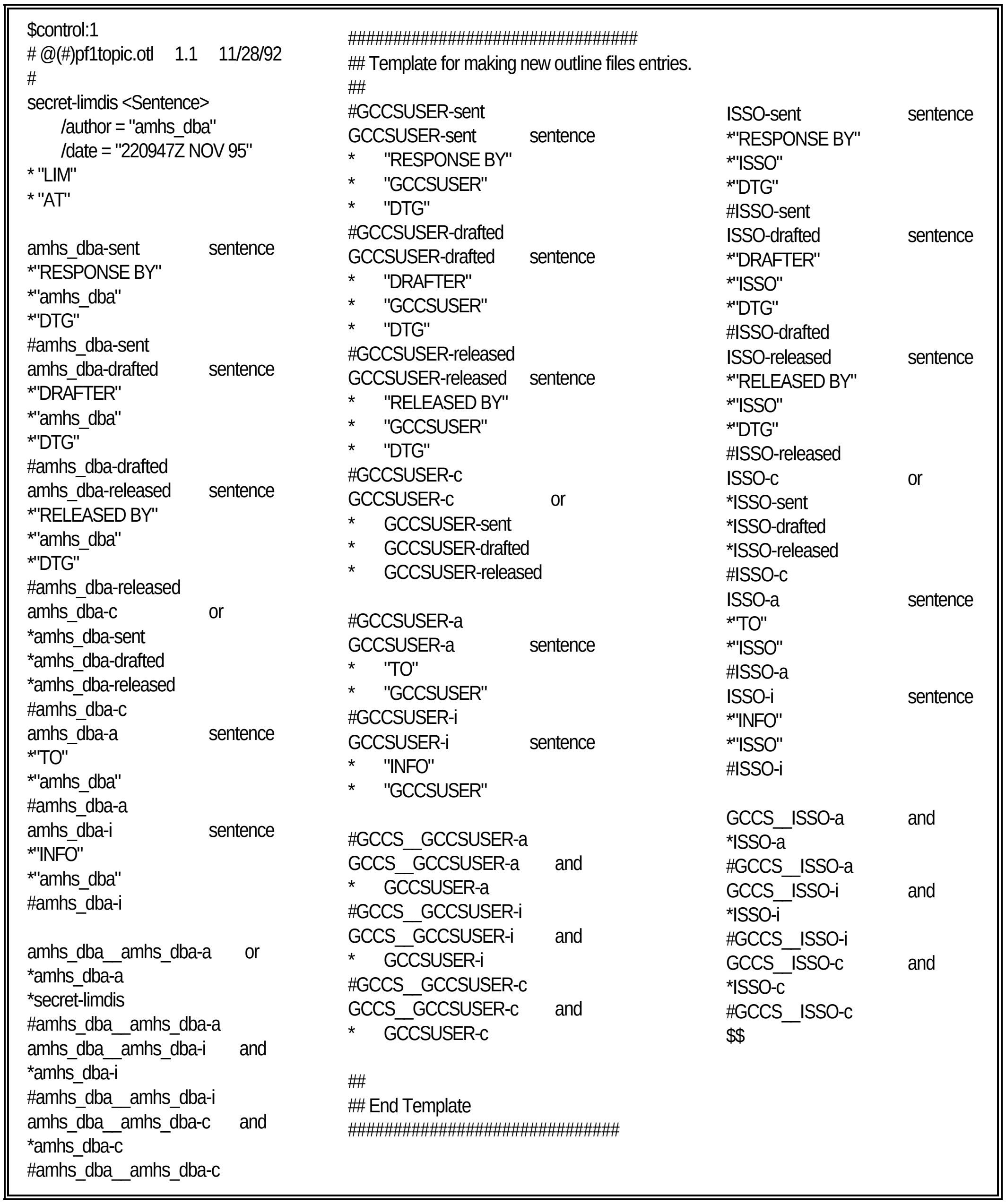
```
$control: 1
profile:
{
   # Begin ACTION amhs_dba
 profile: '<FILTER>(DOCSOURCE = AUTODIN) AND amhs_dba__amhs_dba-a'
 {
  user: amhs_dba .01
    /notify = /h/AMHS/Server/topic/amhs_users/amhs_dba/./ACTION
    /system = "/h/AMHS/Client/progs/profile_hit $PRECEDENCE $DOCSOURCE amhs_dba__. '$DOC_FN' NA '$SUBJECT'"
 } # End ACTION amhs_dba
   # Begin INFO amhs_dba
 profile: '<FILTER>(DOCSOURCE = AUTODIN) AND amhs_dba__amhs_dba-i'
 {
  user: amhs_dba .01
    /notify = /h/AMHS/Server/topic/amhs_users/amhs_dba/./INFO
    /system = "/h/AMHS/Client/progs/profile_hit $PRECEDENCE $DOCSOURCE amhs_dba__. '$DOC_FN' NA '$SUBJECT'"
 } # End INFO amhs_dba
   # Begin MSGS_SENT amhs_dba
 profile: '<FILTER>(DOCSOURCE = COMEBAK) AND amhs_dba__amhs_dba-c'
 {
  user: amhs_dba .01
    /notify = /h/AMHS/Server/topic/amhs_users/amhs_dba/./MSGS_SENT
    /system = "/h/AMHS/Client/progs/profile_hit $PRECEDENCE $DOCSOURCE amhs_dba__. '$DOC_FN' NA '$SUBJECT'"
 } # End MSGS_SENT amhs_dba
   # Begin ACTION GCCSGCCSUSER
 profile: '<FILTER>(DOCSOURCE = AUTODIN) AND GCCS__GCCSUSER-a'
 {
  user: GCCSGCCSUSER .01
    /notify = /h/AMHS/Server/topic/amhs_users/GCCS/GCCSUSER/ACTION
    /system = "/h/AMHS/Client/progs/profile_hit $PRECEDENCE $DOCSOURCE GCCS__GCCSUSER '$DOC_FN' NA '$SUBJECT'"
 } # End ACTION GCCSGCCSUSER
   # Begin INFO GCCSGCCSUSER
 profile: '<FILTER>(DOCSOURCE = AUTODIN) AND GCCS__GCCSUSER-i'
 {
  user: GCCSGCCSUSER .01
    /notify = /h/AMHS/Server/topic/amhs_users/GCCS/GCCSUSER/INFO
    /system = "/h/AMHS/Client/progs/profile_hit $PRECEDENCE $DOCSOURCE GCCS__GCCSUSER '$DOC_FN' NA '$SUBJECT'"
 } # End INFO GCCSGCCSUSER
   # Begin MSGS_SENT GCCSGCCSUSER
 profile: '<FILTER>(DOCSOURCE = COMEBAK) AND GCCS__GCCSUSER-c'
 {
  user: GCCSGCCSUSER .01
    /notify = /h/AMHS/Server/topic/amhs_users/GCCS/GCCSUSER/MSGS_SENT
    /system = "/h/AMHS/Client/progs/profile_hit $PRECEDENCE $DOCSOURCE GCCS__GCCSUSER '$DOC_FN' NA '$SUBJECT'"
 } # End MSGS_SENT GCCSGCCSUSER
   # Begin ACTION willie
 profile: '<FILTER>(DOCSOURCE = Autodin) AND willie_a'
 {
  user: willie .01
    /notify = /h/AMHS/Server/topic/amhs_users/willie/./ACTION
    /system = "/h/AMHS/Client/progs/profile_hit $PRECEDENCE $DOCSOURCE willie__. '$DOC_FN' NA '$SUBJECT'"
 } # End ACTION willie
   # Begin INFO willie
 profile: '<FILTER>(DOCSOURCE = Autodin) AND willie_i'
 {                    additional profiles are appended to the file here by the Queue Manager
```

Figure 2-34. Default Profiles File for pf1

There must be three topics for each user for receiving message traffic profiled to their account.

-a for ACTION: Messages which require action by the user.
-i for INFO: Messages which are of informational interest to the user.
-c for MSGS_SENT: Messages which were released to AUTODIN from GCCS.

In addition to ACTION, INFO, and MSGS_SENT criteria the user may want key words included. Figure 2-38 shows how a typical key word child might look.



```
$control:1
# @(#)pf1topic.otl    1.1    11/28/92
#
secret-limdis <Sentence>
        /author = "amhs_dba"
        /date = "220947Z NOV 95"
* "LIM"
* "AT"

amhs_dba-sent           sentence
*"RESPONSE BY"
*"amhs_dba"
*"DTG"
#amhs_dba-sent
amhs_dba-drafted        sentence
*"DRAFTER"
*"amhs_dba"
*"DTG"
#amhs_dba-drafted
amhs_dba-released       sentence
*"RELEASED BY"
*"amhs_dba"
*"DTG"
#amhs_dba-released
amhs_dba-c              or
*amhs_dba-sent
*amhs_dba-drafted
*amhs_dba-released
#amhs_dba-c
amhs_dba-a              sentence
*"TO"
*"amhs_dba"
#amhs_dba-a
amhs_dba-i              sentence
*"INFO"
*"amhs_dba"
#amhs_dba-i

amhs_dba__amhs_dba-a      or
*amhs_dba-a
*secret-limdis
#amhs_dba__amhs_dba-a
amhs_dba__amhs_dba-i      and
*amhs_dba-i
#amhs_dba__amhs_dba-i
amhs_dba__amhs_dba-c      and
*amhs_dba-c
#amhs_dba__amhs_dba-c

################################
## Template for making new outline files entries.
##
#GCCSUSER-sent
GCCSUSER-sent           sentence
*       "RESPONSE BY"
*       "GCCSUSER"
*       "DTG"
#GCCSUSER-drafted
GCCSUSER-drafted        sentence
*       "DRAFTER"
*       "GCCSUSER"
*       "DTG"
#GCCSUSER-released
GCCSUSER-released       sentence
*       "RELEASED BY"
*       "GCCSUSER"
*       "DTG"
#GCCSUSER-c
GCCSUSER-c                  or
*       GCCSUSER-sent
*       GCCSUSER-drafted
*       GCCSUSER-released

#GCCSUSER-a
GCCSUSER-a              sentence
*       "TO"
*       "GCCSUSER"
#GCCSUSER-i
GCCSUSER-i              sentence
*       "INFO"
*       "GCCSUSER"

#GCCS__GCCSUSER-a
GCCS__GCCSUSER-a        and
*       GCCSUSER-a
#GCCS__GCCSUSER-i
GCCS__GCCSUSER-i        and
*       GCCSUSER-i
#GCCS__GCCSUSER-c
GCCS__GCCSUSER-c        and
*       GCCSUSER-c

##
## End Template
##############################

ISSO-sent               sentence
*"RESPONSE BY"
*"ISSO"
*"DTG"
#ISSO-sent
ISSO-drafted            sentence
*"DRAFTER"
*"ISSO"
*"DTG"
#ISSO-drafted
ISSO-released           sentence
*"RELEASED BY"
*"ISSO"
*"DTG"
#ISSO-released
ISSO-c                  or
*ISSO-sent
*ISSO-drafted
*ISSO-released
#ISSO-c
ISSO-a                  sentence
*"TO"
*"ISSO"
#ISSO-a
ISSO-i                  sentence
*"INFO"
*"ISSO"
#ISSO-i

GCCS__ISSO-a            and
*ISSO-a
#GCCS__ISSO-a
GCCS__ISSO-i            and
*ISSO-i
#GCCS__ISSO-i
GCCS__ISSO-c            and
*ISSO-c
#GCCS__ISSO-c
$$
```

Figure 2-35. Topic Outline Profiles File

The AMHS System Administrator develops profiling criteria consistent with the user's needs, using the TOPIC Editor window. Each topic begins with a name, typically the user's Project/Position Pair (Figure 2-37), e.g. **GCCS_CURROPS -a**, appended with **-a** for ACTION, **-i** for INFO, or **-c** for MSGS_SENT comeback copy. It is important to note that the user will use the same process locally to build libraries of retrospective search criteria (profiles) to review the message database. Following are criteria building strategies for these queues.

2.4.1.1 Action Queue

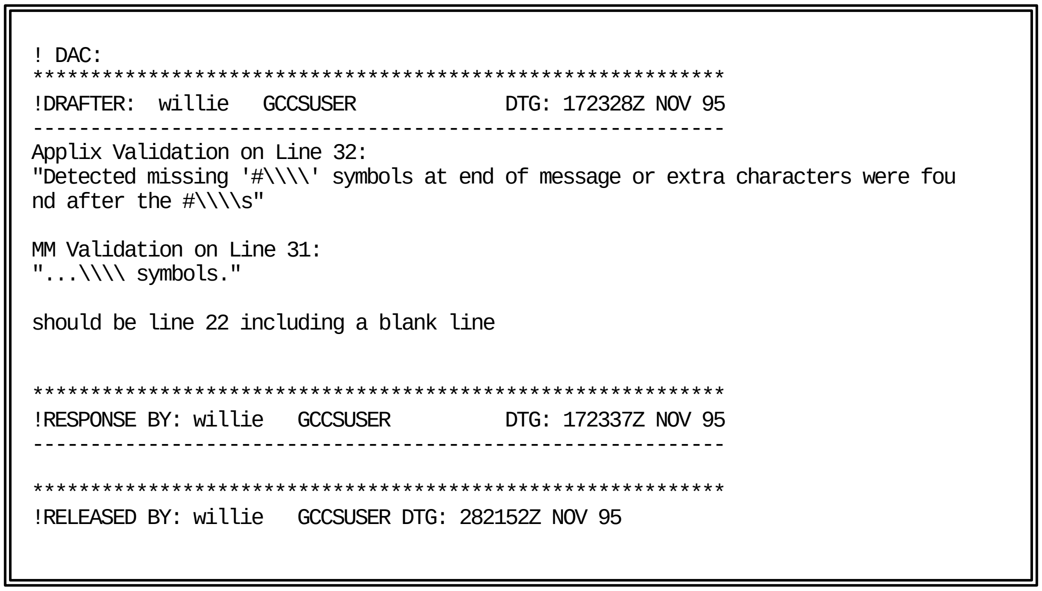
The typical approach is to construct a topic which identifies the AUTODIN messages which contain the user's office symbol in the “TO” field. The topic optionally also identifies desired keywords within the message text. Figure 2-38 shows an example of how a “TO” child might be constructed.

2.4.1.2 Info Queue

The typical approach is to construct a topic which identifies the AUTODIN messages that contain the user's office symbol in the “INFO” field. The topic optionally identifies desired keywords within the message text. Messages which meet the ACTION criteria for a user can be excluded from that user's INFO queue by concatenating the user’s ACTION topic with a NOT operator to the INFO queue topic profile.

2.4.1.3 MSGS_SENT Queue

The typical strategy is to construct a topic which keys off the “RELEASED BY” field of the Message Action Record (MAR) of the message. See Figure 2-36. The topic can specify all messages released through GCCS or just the messages released by a certain user. The MAR files are located at **/h/AMHS/Server/topic/amh_db/coord/comeback/dxxxxxx.mar**.



```
! DAC:
************************************************************
!DRAFTER:  willie   GCCSUSER             DTG: 172328Z NOV 95
------------------------------------------------------------
Applix Validation on Line 32:
"Detected missing '#\\\\' symbols at end of message or extra characters were fou
nd after the #\\\\s"

MM Validation on Line 31:
"...\\\\ symbols."

should be line 22 including a blank line


************************************************************
!RESPONSE BY: willie   GCCSUSER          DTG: 172337Z NOV 95
------------------------------------------------------------

************************************************************
!RELEASED BY: willie   GCCSUSER DTG: 282152Z NOV 95
```

Figure 2-36. Typical MAR File

MAR files contain routing information from the Message Manager buckslip, including comments. The users can have profile criteria built by user name, project or position as Drafter, Reviewer, or Releaser, and even note text in the body if they want to flag a message for CBC filing during the buckslip process. Users typically fall in the following categories:

(1) All messages they released.

(2) All messages they released as Drafter or Coordinator or Release Authority.

(3) Every message released from the organization.

Whatever the requirement profiling, the MAR files will be able to identify and route the CBC to the user's MSGS_SENT queue.

NOTE: The DAC process in the **sat_feed** attaches the security groups to each message. The users that do not match the DAC security requirement will not have access to messages profiled by the TOPIC Real-Time Processor. The profiling process is a two-step process: first, by the **sat_feed** for security; and second by the TOPIC Processor for address and content.

2.4.2 TOPIC Editor

The TOPIC Editor allows you to view topics one at a time in full detail. This method shows words, operators, and weights included in the topic.

Topics shown in the TOPIC Editor can be displayed in tree format or outline format. The default is tree format. You can modify existing topics or create new ones from the TOPIC Editor.

The TOPIC Editor allows on-line retrospective retrievals to be performed against a given topic at any time. The Administrator can easily identify and correct mistakes/typos. Promotes stepwise refinement of message profiles.

At this point the System Administrator should be familiar with the topic profile editing process. This information is covered in the Verity Inc. document titled TOPIC Database Administrator's Guide V3.1, Vol. 1, under reference number NA-TOP-01-01 in the DISA document library.

Below are pointers to the referenced sections, quoted from the Verity document.

Section 3 - Knowledgebase Preparation[*]

Chapter 8 Topic Trees

8 Topic Trees*

This chapter discusses the elements which comprise *topics*, or groups of related subjects, phrases, and/or words; and their related operators, modifiers, and weights, which are used to perform retrievals when using the TOPIC Retrieval Client for your system interface.

The following topic tree aspects are covered in this chapter:

- Understanding the basic elements and properties which comprise a topic
- Understanding the relationship between topics which have been defined in the topic tree
- Understanding how to use operators, modifiers, and weights, and how they affect your retrieval results

Chapter 9 Designing Topics*

9 Designing Topics*

In planning a topic set design, the following activities are performed:

- Defining the retrieval needs of your organization
- Understanding how topics will be used by TOPIC users
- Identifying those individuals who provide the topic design requirements and/or build topics
- Implementing the design strategies that will incorporate topics into sets

This chapter discusses the activities and methodologies which can be used to design topic sets which most effectively meet your retrieval needs.

The other sections of the TOPIC Database document are for information only. The processes required to configure, operate and maintain the AMHS are handled through the system administration tools and the **topic_cmd** script. The instructions under Topic Weights (starting Page 8-46) and Topic Scoring (starting Page 8-51) are for use with topics built by users for retrospective searches, when they are looking for types of information.

(1) The definition of a topic is a collection of words and phrases joined by “logical operators” which describe concept, subject or idea.

High_Precedence <or>	EUCOM AOR Regions <any>
* Precedence: Y	* Eastern Europe
* Precedence: Z	* Western Europe
* Precedence: O	* African
	* Adriatic

The precedence can be read from the first character of Format Line 4 of a JANAP 128 Header format message.

(2) Logical operators are used to specify relationships between words, topics, or both. You assign operators to control how documents are retrieved. The most commonly used operators at AMHS sites:

<AND>
<NOT>
<OR>
<ALL>
<Phrase>
<ANY>
<Sentence>

The TOPIC Editor makes editing easy with the Handle Icon. The Handle Icon allows you to select and move nodes and their connected children with drag and drop. The handle appears to the left of each node. With complicated multilevel trees, it is difficult to see the whole tree in one window. The Collapsing and Expanding Nodes feature allows you to close or collapse children and levels to focus on the areas you are editing. To expand or collapse a node, click on the icon to the right of the node.

2.4.3 Creating Topics

The TOPIC Editor must be launched specifically as your user before creating a topic. The TOPIC Editor is easy to use, and the commands are intuitive if you understand the principles outlined in the TOPIC manuals. The following steps will launch TOPIC Editor and create a topic:

(1) Open an Xterm window from your launch window and under your individual GCCS desktop account as the **amhs_dba** using the following commands:

(a) **xhost +**

(b) **su - amhs_dba**

(c) Password for **amhs_dba** account.

(d) **runclient.**

(2) Choose the operator to be used with the parent topic you are defining.

(3) Enter the topic name.

(4) Create branch topic and leaf topic children, assigning operators and naming the branch topics and leaf topics as they are created.

(5) Assign the weights to the branch topic and leaf topic children.

(6) Assign modifiers, if desired.

The following is an example of a typical message profile request from a user at a major command. An outline of the three main steps taken by the AMHS Administrator to implement this request is described below:

(1) An AMHS TOPIC user account must be opened with the Account Manager Sys Admin Tool.

(2) The message routing content (profile) must be created using the TOPIC Editor.

(3) The profile must be connected to the user's queue with the Queue Manager Sys Admin Tool.

User Requirement Maj. Smith works in the ECJ3 Current Operations shop at Headquarters, European Command (HQ EUCOM). The GCCS account has been created and Maj. Smith is profiled into Project GCCS and Position CURROPS. Maj. Smith indicates that their office symbol is ECJ33 or ECJ3ODO, and that the keywords of interest are REVOLUTION, CONTINGENCY, and 'MSGID OPORD'.

(1) Add the AMHS account based on Maj. Smith's Project/Position Pair using the Account Manager tool.

(2) Run the AMHS Client as the **amhs_dba** using the commands described in the beginning of this section.

(a) Select File -> New Query.

(b) Select Query-> Query Type-> Topic Query.

(c) Resize the window to allow editing within the TOPIC Editor Window.

(d) Construct the Topic. See Figure 2-37.

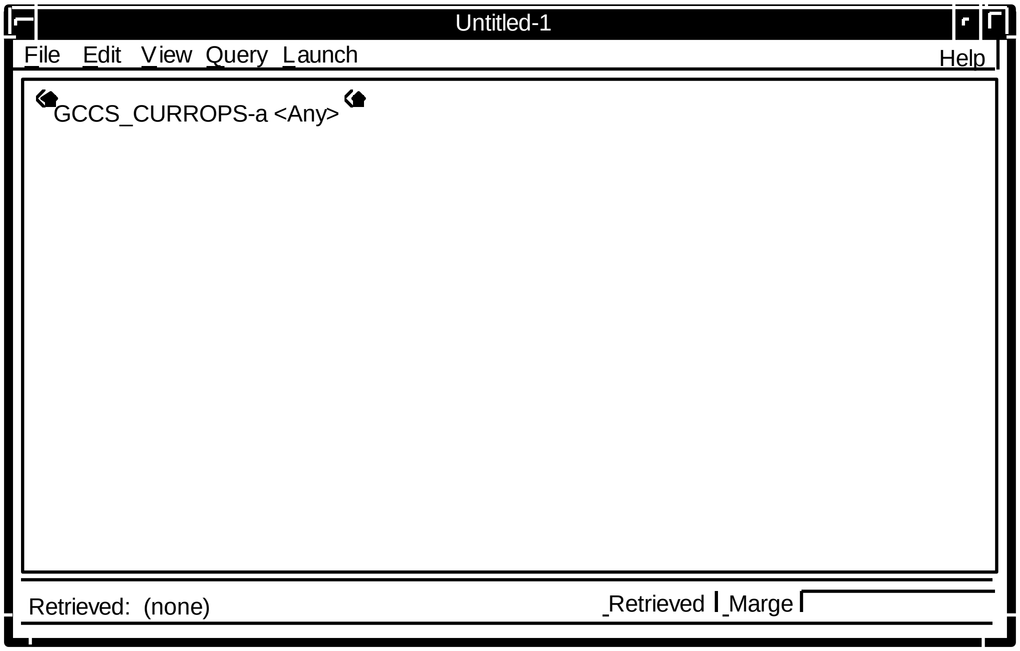


Figure 2-37. GCCS_CURROPS Topic

(3) Typical steps to construct a topic profile criteria using the TOPIC Editor include the following:

(a) Create a parent topic. Incorporate the user's Project/Position pair in the name.

(b) Use the **Query -> Add Operator** menu option to create a two subtopics subordinate to the parent topic.

(c) Create a template for the user's “TO” criteria. It consists of three parts:

1) The “TO” keyword.
2) The site's PLAs
3) The user's office symbols.

The three components are logically joined by the “Sentence” operator.

(d) Use the **Query -> Add Child** menu option to fill in the specific “TO” criteria for the user.

(e) Fill in the "Keyword” criteria at the lower subtopic level.

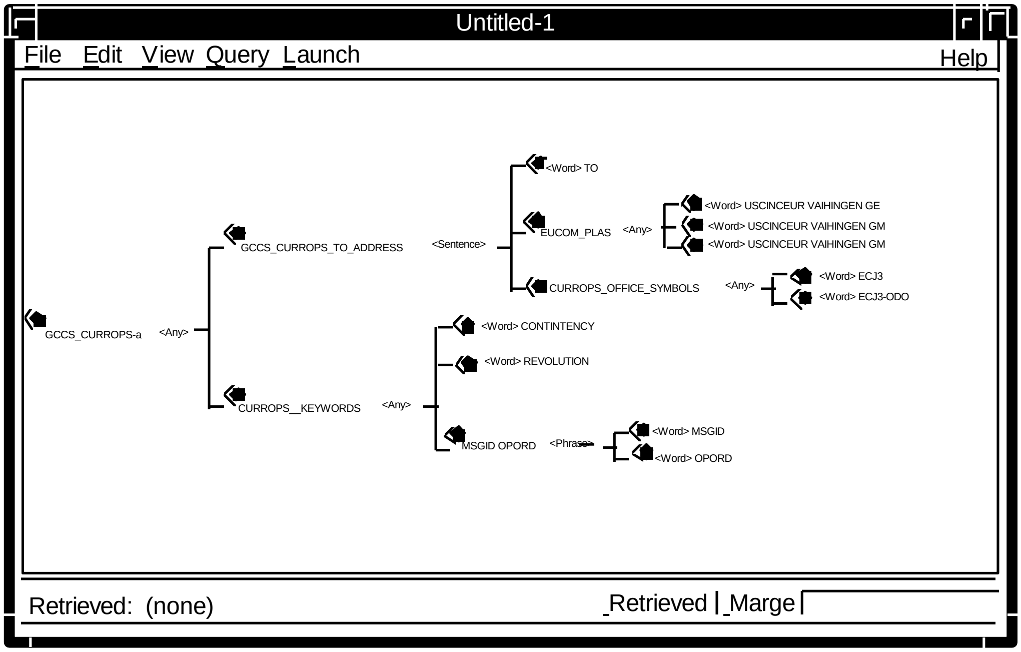
(f) Once complete, the topic can be tested against the existing database by clicking on the **Retrieve** button.

1) Verify that the retrieval returns the expected results.
2) Make sure you understand any variances from the expected results before proceeding.

(g) Save the topics by selecting **File -> Save Topics**. Remember what you named the topic. Section 2.4.4 gives you some naming concepts.

At this point you must make the new profile known to the Central Profiler. This is handled through the System Admin Queue Manager tool. In Section 5, the Queue Manager processes are described, and how to run the **mkusrtop** program to complete the profiling process. See Figure 2-38.

NOTE: This is an important step since it takes a snapshot of the current profile criteria set. The profile criteria set can be reloaded into the AMHS from this outline file if the active profile criteria set ever becomes corrupted.



Retrieved: (none)

Figure 2-38. Typical Topic Profile

2.4.4 Message Routing Recommendations

Developing a knowledge base topic tree, with close interaction with the user, is a good policy, and developing a standard topic naming method is essential. The goal is to be able to recognize a topic by its name, not by its underlying criteria. After you have several hundred topics on file, keeping track of them can be difficult.

The TOPIC Editor provides the ability to perform retrievals of your topic as it is being built. This is an invaluable tool for checking the correctness of your topic. Saved topics can be reused by many parent topics. Creating topics which are modular can reduce the topic building process considerably. It is essential that the profile criteria set be saved in outline format whenever an update is performed. This allows for recovery in case of unexpected system anomalies. You should keep a separate copy of each topic along with your notes that were used to develop the topic, and a set of test messages, used to test the profile, for future debugging and maintenance. If you use a modular design of the topic and/or subtopics, they can be made available to the users for retrospective searches of existing and restored messages.

Topic is a powerful and complex tool with many more features than the average site will need for normal day-to-day message traffic distribution. The tools and procedures supplied with the AMHS package will make the routine tasks relatively easy. The downside of the tools and methods is that if you decide to explore some of the more complex features of TOPIC without understanding how the tools automate some of the processes you can get the system out of sync. This is especially true of Account Manager and Queue Manager configurations described in Section 5.2.

Develop a site-specific demo suite to show your users to assist in their understanding of the retrospective search process. Also, include a discussion of message format beyond the Line 12 body text. Use Figure 2-47, Typical AUTODIN Message, in this Section as a baseline. This will help to create a common language to assist in the knowledge base topic design process.

This process and the creation of a dead letter user account with all the parent topic attach at <NOT> operators, and the use of the Processing Undelivered Message process described in Section 4.3 will significantly reduce your support work load.

2.5 OUTBOUND MESSAGE PROCESSING

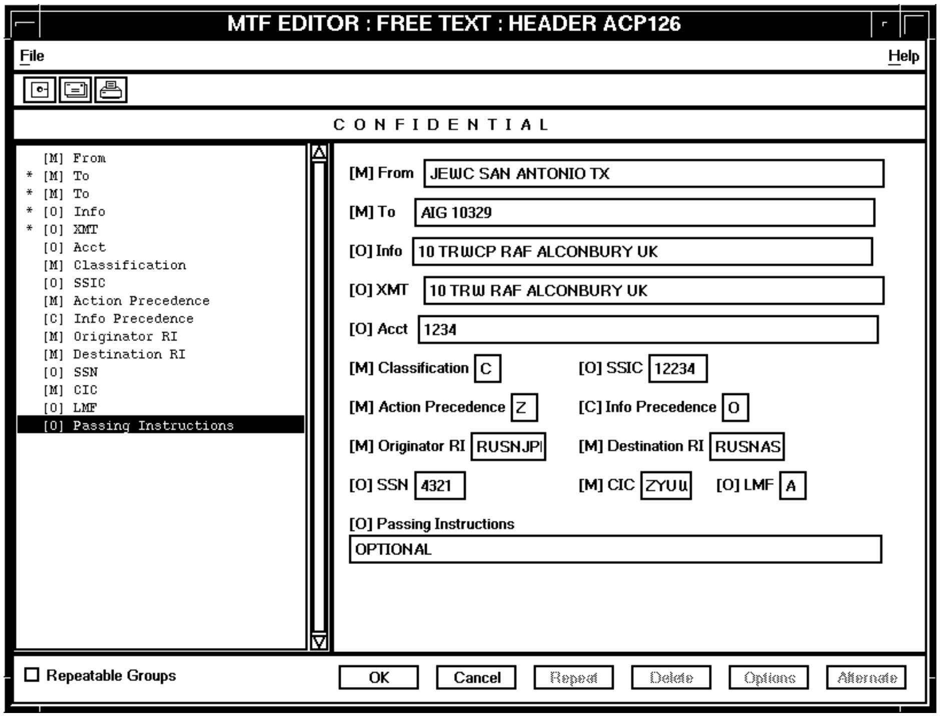
AUTODIN messages may be authored with the assistance of the MTF Editor or created/edited directly in Applix Words, but can only be released from the Message Manager Buckslip Window or special applications that link to the release APIs by persons with release authorization. The validation tools can assist the review process prior to release.

In special cases it may be necessary to release a message that is not consistent with the validation criteria. This flexibility is allowed by selecting Validate on Release [NO] in the Message Manager Validation screen prior to release. It is important to have run validation and determined that all other criteria have been met and the specific error is consistent with the message to be released. A clear understanding of how the validation process works and what is checked against what criteria will assist in making good decisions about the format and content of the special message to be released.

Section 2.5.1 gives a basic overview of the message authoring and release process. Section 2.5.2 addresses the validation and release criteria, and Section 2.5.3 describes the files and tables used to configure and customize the system.

2.5.1 Message Authoring, Processing And Releasing

Messages may be originated in JANAP 128, ACP 126, or DD173 format.



```
  [M] From
* [M] To
* [M] To
* [O] Info
* [O] XMT
  [O] Acct
  [M] Classification
  [O] SSIC
  [M] Action Precedence
  [C] Info Precedence
  [M] Originator RI
  [M] Destination RI
  [O] SSN
  [M] CIC
  [O] LMF
  [O] Passing Instructions
```

[M] From JEWC SAN ANTONIO TX

[M] To AIG 10329

[O] Info 10 TRWCP RAF ALCONBURY UK

[O] XMT 10 TRW RAF ALCONBURY UK

[O] Acct 1234

[M] Classification C [O] SSIC 12234

[M] Action Precedence Z [C] Info Precedence O

[M] Originator RI RUSNJPI [M] Destination RI RUSNAS

[O] SSN 4321 [M] CIC ZYUU [O] LMF A

[O] Passing Instructions OPTIONAL

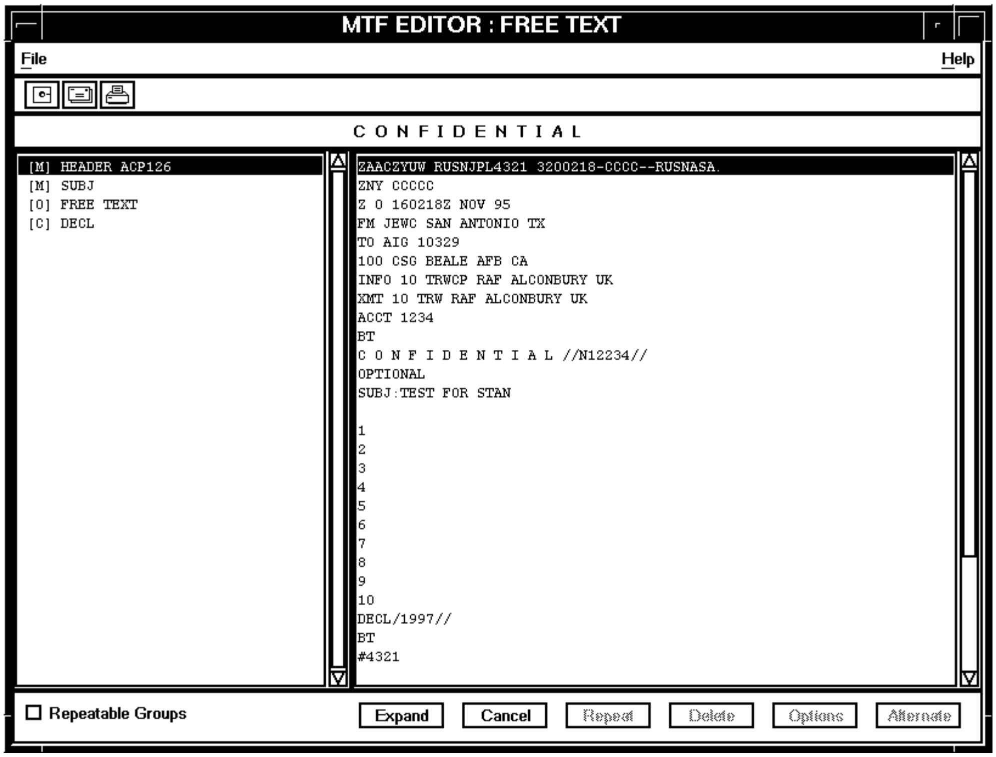
☐ Repeatable Groups

Figure 2-39. MTF Editor Screen 1

The MTF Editor Screen 1 (Figure 2-39) uses the MAST__PLA.CCA file to assist in selecting approved To/From and Info/XMT addresses and the other boxes have character set limits to accept appropriate responses. By highlighting a selection on the left and clicking the Options button at the bottom right, more choices are made available.

The Originating Station Routing Indicator (OSRI) and Destination Station Routing Indicator (DSRI) are supplied by the Ri.CCA file and are site/switch-specific. The MAST_PLA.CCA and Ri.CCA files can be edited with any standard ASCII text editor such as vi from an Xterm window.

MTF Editor Screen 2 (Figure 2-40) assists in completing the message.



```
[M] HEADER ACP126
[M] SUBJ
[O] FREE TEXT
[C] DECL
```

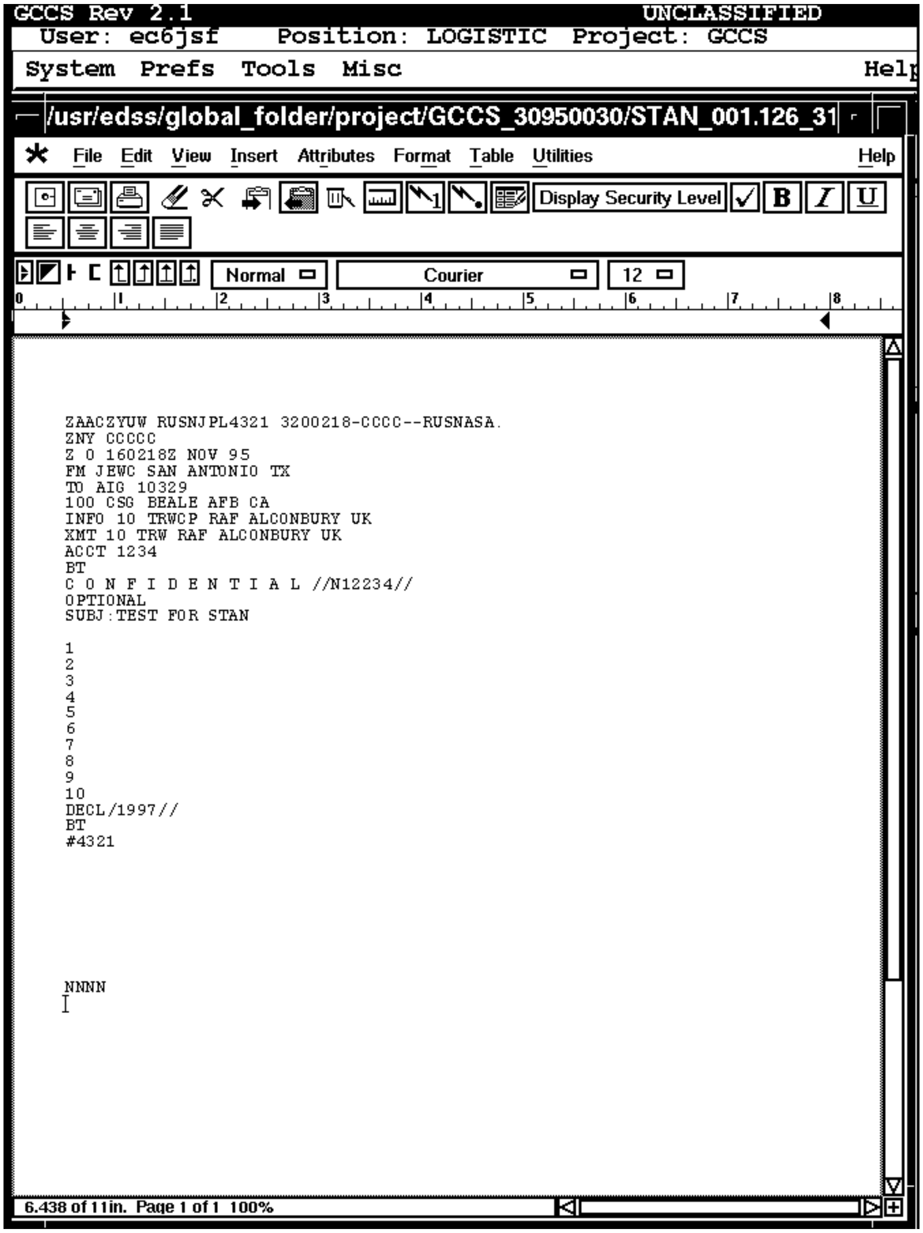
```
ZAACZYUW RUSNJPL4321 3200218-CCCC--RUSNASA.
ZNY CCCCC
Z O 160218Z NOV 95
FM JEWC SAN ANTONIO TX
TO AIG 10329
100 CSG BEALE AFB CA
INFO 10 TRWCP RAF ALCONBURY UK
XMT 10 TRW RAF ALCONBURY UK
ACCT 1234
BT
C O N F I D E N T I A L //N12234//
OPTIONAL
SUBJ:TEST FOR STAN

1
2
3
4
5
6
7
8
9
10
DECL/1997//
BT
#4321
```

Figure 2-40. MTF Editor Screen 2

For messages that are regularly sent to the same destination, it may be faster to use the Applix Words Editor (Figure 2-41) to edit a message template created with the MTF Editor. In either case, preliminary validation should be done by using Save_Validate from MTF Editor or the check button **[✓]** on the Applix Words Editor button toolbar (Figure 2-41). The Validation errors are reported by line number and description to assist in making corrections. (Figures 2-42 and 2-46.)

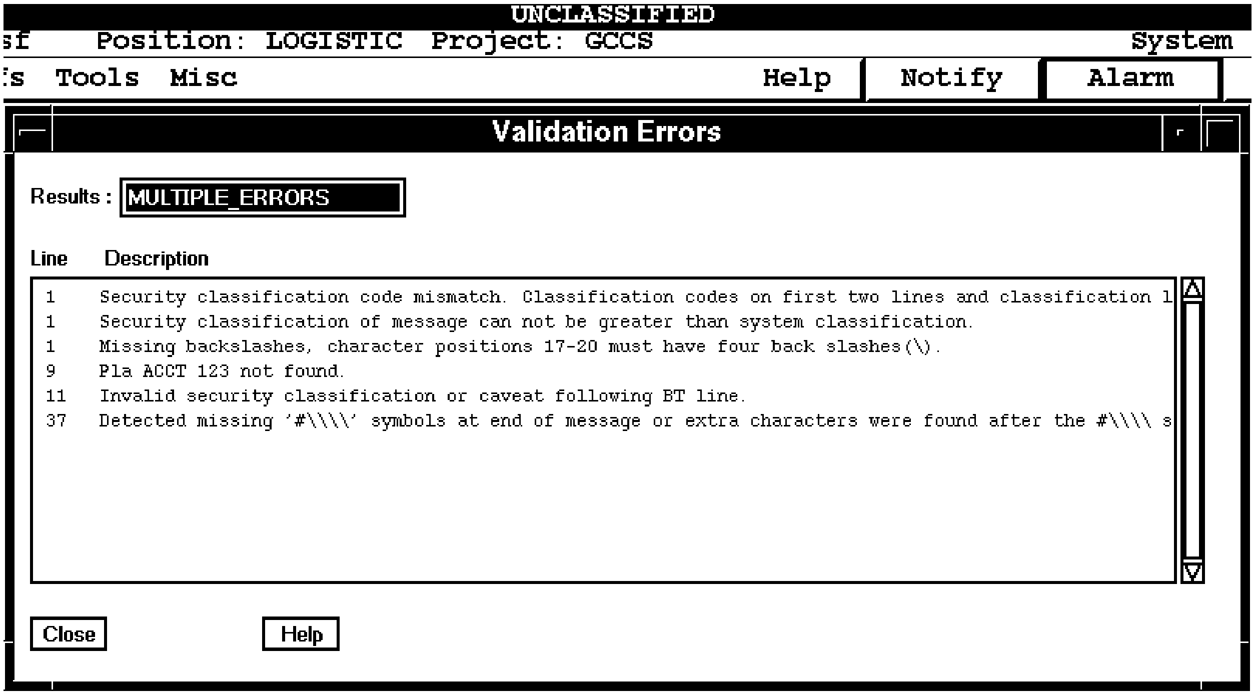
In the special case of the USMTF messages, the MTF Editor accepts only USMTF body text-formatted messages and carefully checks the format of any changes. These messages should be edited only with the MTF Editor. The validation process checks the Header/To/From information and is the same process for these messages as for plain language messages (free text). Messages are stored in the Desktop Foldering System, which can be viewed by Message Manager: Main Window Folder (Figure 2-43), awaiting further processing. The message is attached to a Buckslip for approval routing and release (Figure 2-44). Prior to release a final validation is performed (Figure 2-46).

Messages can also be created and/or edited using the Applix Words Editor by authoring new or modifying existing messages and using the built-in validation tool **[✓]**. Any message created with the MTF Editor and subsequently edited using Applix is not available as an MTF-editable template, i.e., cannot be edited using the MTF Editor application.



```
ZAACZYUW RUSNJPL4321 3200218-CCCC--RUSNASA.
ZNY CCCCC
Z 0 160218Z NOV 95
FM JEWC SAN ANTONIO TX
TO AIG 10329
100 CSG BEALE AFB CA
INFO 10 TRWCP RAF ALCONBURY UK
XMT 10 TRW RAF ALCONBURY UK
ACCT 1234
BT
C O N F I D E N T I A L //N12234//
OPTIONAL
SUBJ:TEST FOR STAN

1
2
3
4
5
6
7
8
9
10
DECL/1997//
BT
#4321






NNNN
```

Figure 2-41. Applix Words Editor

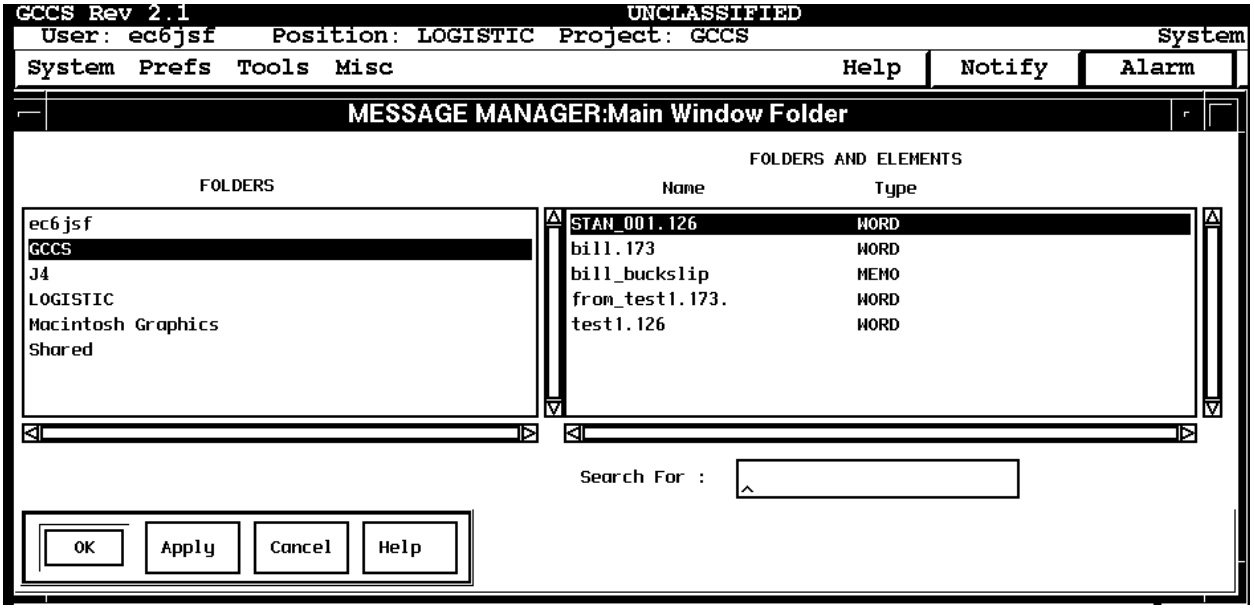

UNCLASSIFIED
sf Position: LOGISTIC Project: GCCS System
fs Tools Misc Help Notify Alarm
Validation Errors
Close
Help

Results : MULTIPLE_ERRORS

```
Line  Description
1     Security classification code mismatch. Classification codes on first two lines and classification l
1     Security classification of message can not be greater than system classification.
1     Missing backslashes, character positions 17-20 must have four back slashes(\).
9     Pla ACCT 123 not found.
11    Invalid security classification or caveat following BT line.
37    Detected missing '#\\\\' symbols at end of message or extra characters were found after the #\\\\ s
```

Figure 2-42. MTF/Applix Words Validation Errors Report

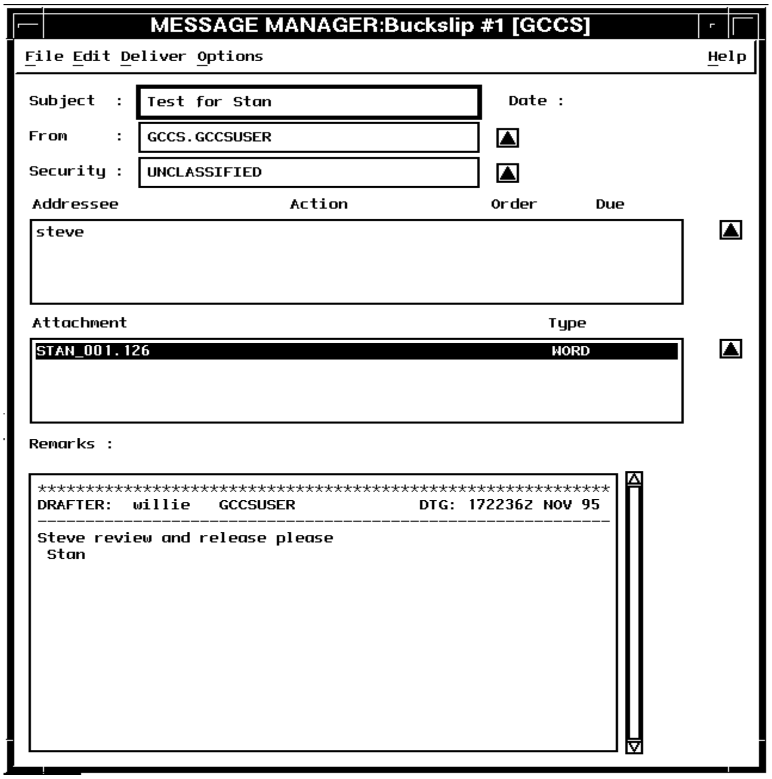
The Message Manager Main Window folder gives the user easy access to scan folders for messages that might be template candidates.


GCCS Rev 2.1 UNCLASSIFIED
User: ec6jsf Position: LOGISTIC Project: GCCS System
System Prefs Tools Misc Help Notify Alarm
MESSAGE MANAGER:Main Window Folder
FOLDERS
FOLDERS AND ELEMENTS
Search For :
OK
Apply
Cancel
Help

FOLDERS:
- ec6jsf
- GCCS
- J4
- LOGISTIC
- Macintosh Graphics
- Shared

Name	Type
STAN_001.126	WORD
bill.173	WORD
bill_buckslip	MEMO
from_test1.173.	WORD
test1.126	WORD

Figure 2-43. Message Manager



Subject : Test for Stan

Date :

From : GCCS.GCCSUSER

Security : UNCLASSIFIED

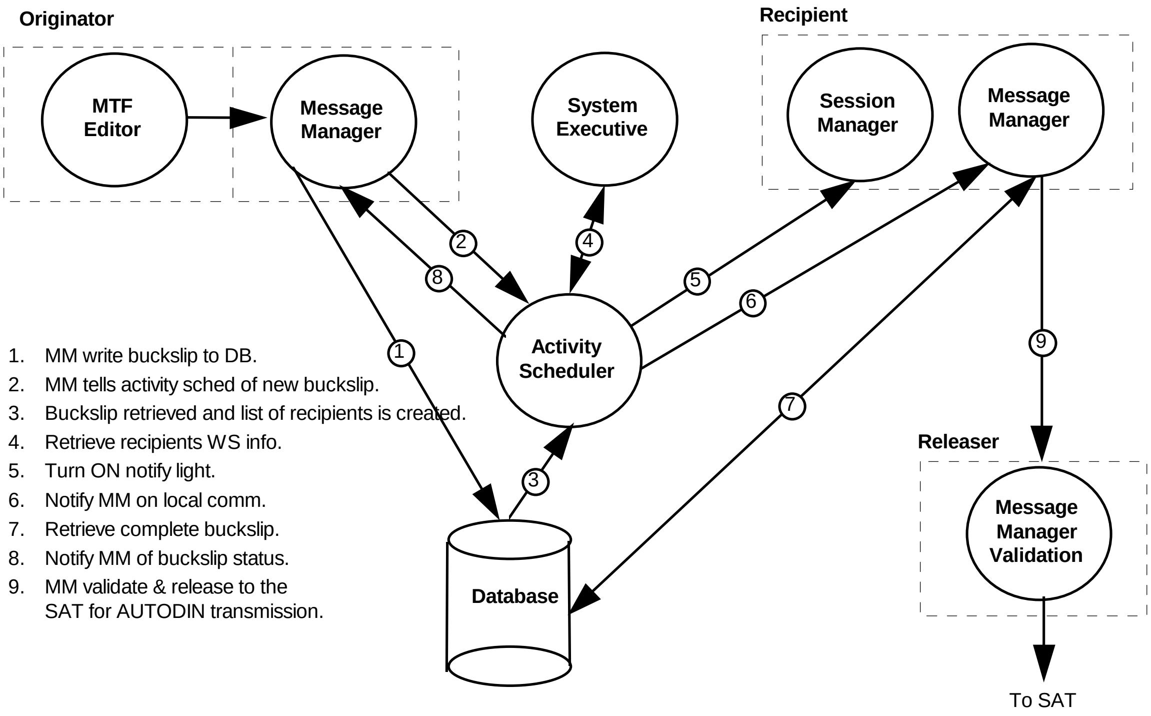
Addressee	Action	Order	Due
steve			

Attachment	Type
STAN_001.126	WORD

Remarks :

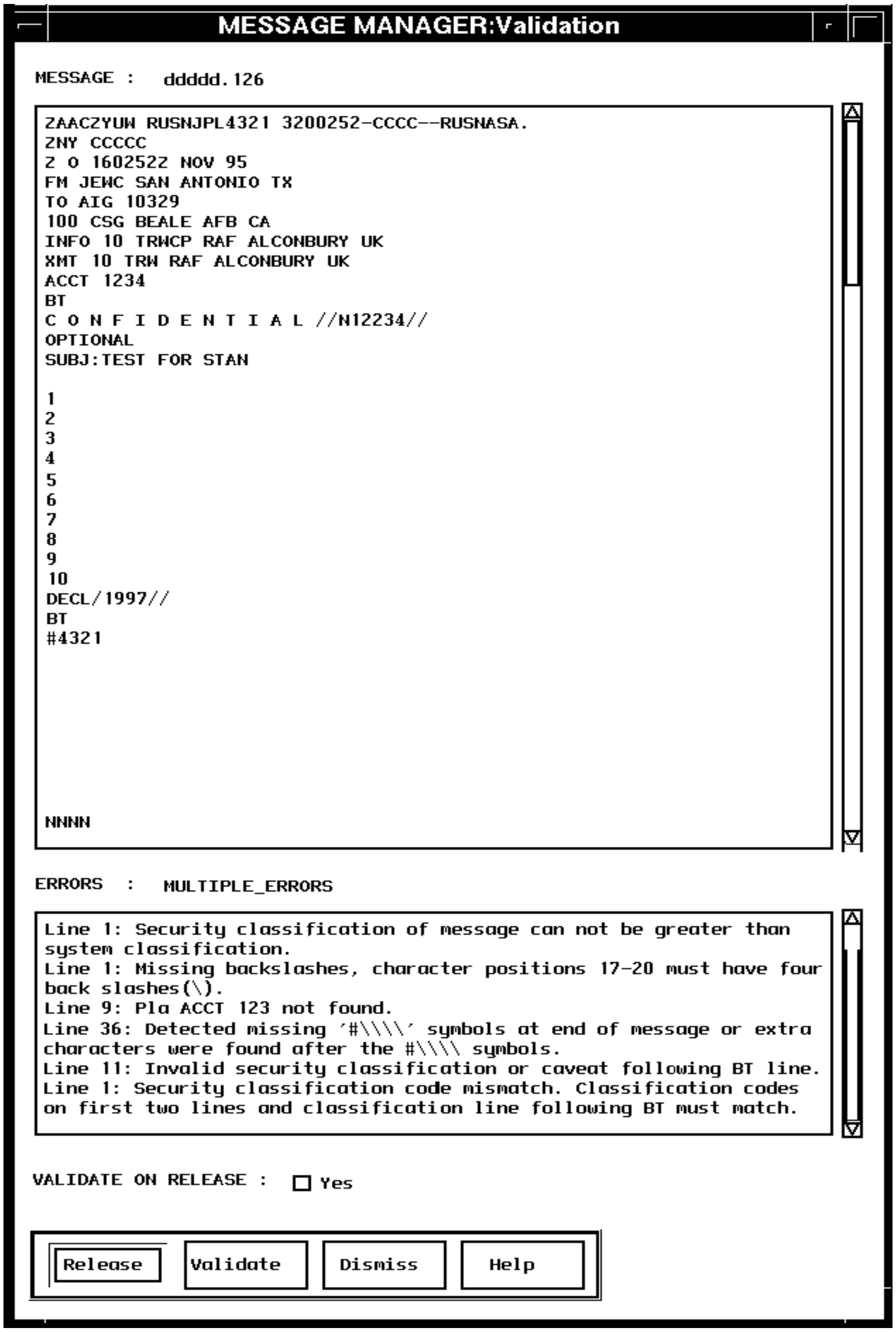
```
*************************************************************
DRAFTER:  willie   GCCSUSER            DTG: 172236Z NOV 95
-------------------------------------------------------------
Steve review and release please
 Stan
```

Figure 2-44. Buckslip



1. MM write buckslip to DB.
2. MM tells activity sched of new buckslip.
3. Buckslip retrieved and list of recipients is created.
4. Retrieve recipients WS info.
5. Turn ON notify light.
6. Notify MM on local comm.
7. Retrieve complete buckslip.
8. Notify MM of buckslip status.
9. MM validate & release to the SAT for AUTODIN transmission.

Figure 2-45. Buckslip Data Flow



MESSAGE : ddddd.126

```
ZAACZYUW RUSNJPL4321 3200252-CCCC--RUSNASA.
ZNY CCCCC
Z O 160252Z NOV 95
FM JEWC SAN ANTONIO TX
TO AIG 10329
100 CSG BEALE AFB CA
INFO 10 TRWCP RAF ALCONBURY UK
XMT 10 TRW RAF ALCONBURY UK
ACCT 1234
BT
C O N F I D E N T I A L //N12234//
OPTIONAL
SUBJ:TEST FOR STAN

1
2
3
4
5
6
7
8
9
10
DECL/1997//
BT
#4321








NNNN
```

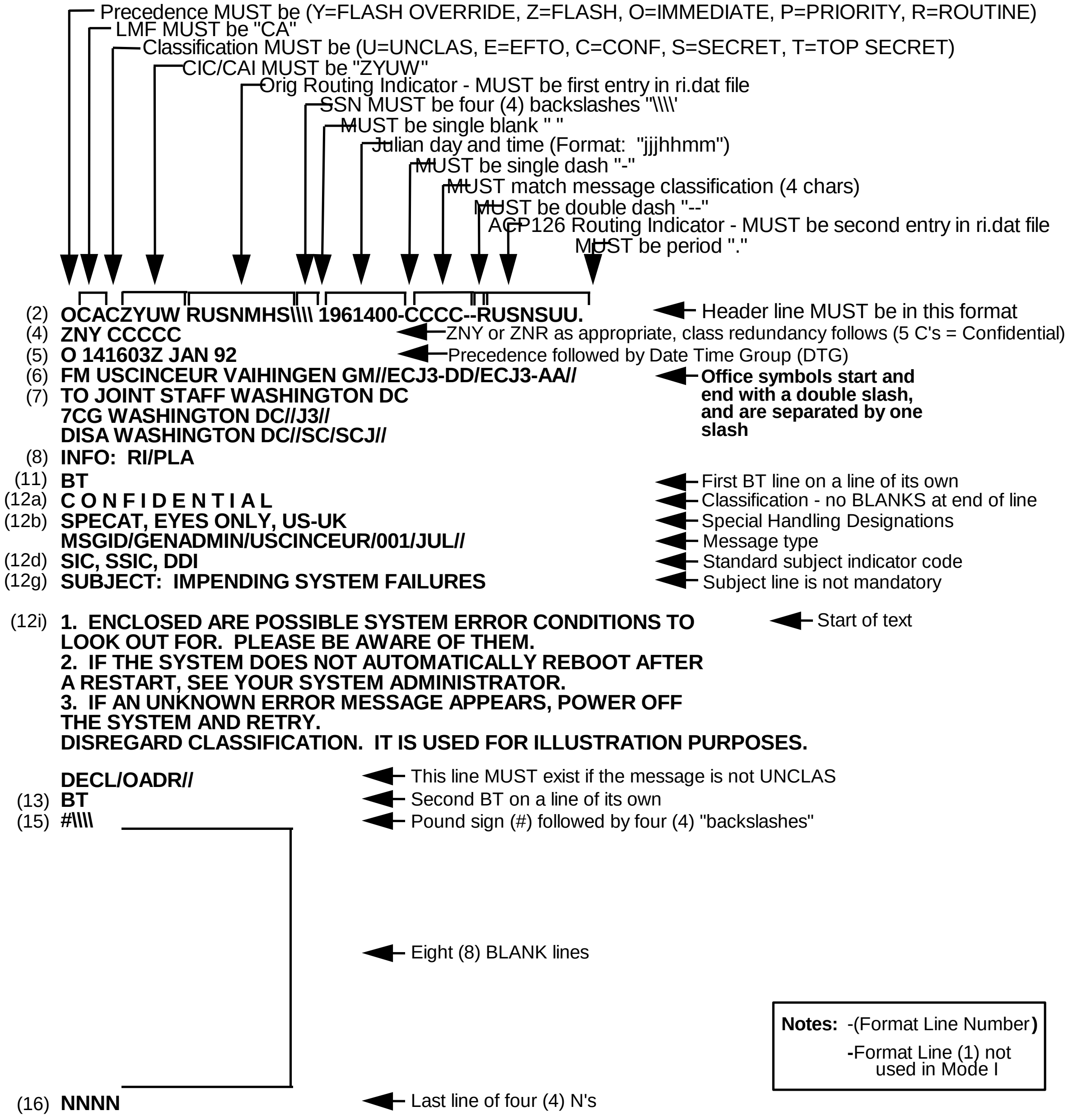
ERRORS : MULTIPLE_ERRORS

```
Line 1: Security classification of message can not be greater than
system classification.
Line 1: Missing backslashes, character positions 17-20 must have four
back slashes(\).
Line 9: Pla ACCT 123 not found.
Line 36: Detected missing '#\\\\' symbols at end of message or extra
characters were found after the #\\\\ symbols.
Line 11: Invalid security classification or caveat following BT line.
Line 1: Security classification code mismatch. Classification codes
on first two lines and classification line following BT must match.
```

VALIDATE ON RELEASE : □ Yes

Figure 2-46. Message Manager Validation Window

When messages meet all the validation criteria the message may be released for AUTODIN delivery. In special cases it may be necessary to release a message in a format inconsistent with the validation process. Section 2.5.2 describes the validation process to assist in making this decision. If it is determined that it is appropriate to release the message with a validation error, select Validate on Release [NO] (the Yes/No is a toggle) after the review process has determined that the message is acceptable for release and any outstanding validation errors are not inconsistent with the site security policies.


Precedence MUST be (Y=FLASH OVERRIDE, Z=FLASH, O=IMMEDIATE, P=PRIORITY, R=ROUTINE)
LMF MUST be "CA"
Classification MUST be (U=UNCLAS, E=EFTO, C=CONF, S=SECRET, T=TOP SECRET)
CIC/CAI MUST be "ZYUW"
Orig Routing Indicator - MUST be first entry in ri.dat file
SSN MUST be four (4) backslashes "\\\\'
MUST be single blank " "
Julian day and time (Format: "jjjhhmm")
MUST be single dash "-"
MUST match message classification (4 chars)
MUST be double dash "--"
ACP126 Routing Indicator - MUST be second entry in ri.dat file
MUST be period "."
Header line MUST be in this format
ZNY or ZNR as appropriate, class redundancy follows (5 C's = Confidential)
Precedence followed by Date Time Group (DTG)
Office symbols start and end with a double slash, and are separated by one slash
First BT line on a line of its own
Classification - no BLANKS at end of line
Special Handling Designations
Message type
Standard subject indicator code
Subject line is not mandatory
Start of text
This line MUST exist if the message is not UNCLAS
Second BT on a line of its own
Pound sign (#) followed by four (4) "backslashes"
Eight (8) BLANK lines
Last line of four (4) N's
Notes: -(Format Line Number**)**
-Format Line (1) not used in Mode I

```
(2)   OCACZYUW RUSNMHS\\\\ 1961400-CCCC--RUSNSUU.
(4)   ZNY CCCCC
(5)   O 141603Z JAN 92
(6)   FM USCINCEUR VAIHINGEN GM//ECJ3-DD/ECJ3-AA//
(7)   TO JOINT STAFF WASHINGTON DC
      7CG WASHINGTON DC//J3//
      DISA WASHINGTON DC//SC/SCJ//
(8)   INFO: RI/PLA
(11)  BT
(12a) C O N F I D E N T I A L
(12b) SPECAT, EYES ONLY, US-UK
      MSGID/GENADMIN/USCINCEUR/001/JUL//
(12d) SIC, SSIC, DDI
(12g) SUBJECT: IMPENDING SYSTEM FAILURES

(12i) 1. ENCLOSED ARE POSSIBLE SYSTEM ERROR CONDITIONS TO
      LOOK OUT FOR. PLEASE BE AWARE OF THEM.
      2. IF THE SYSTEM DOES NOT AUTOMATICALLY REBOOT AFTER
      A RESTART, SEE YOUR SYSTEM ADMINISTRATOR.
      3. IF AN UNKNOWN ERROR MESSAGE APPEARS, POWER OFF
      THE SYSTEM AND RETRY.
      DISREGARD CLASSIFICATION. IT IS USED FOR ILLUSTRATION PURPOSES.

      DECL/OADR//
(13)  BT
(15)  #\\\\

(16)  NNNN
```

Figure 2-47. Typical AUTODIN Message

The format may vary slightly between JANAP 128 and ACP 126, but the basic format issues are described in Figure 2-47. The validation criteria for these messages are described below and illustrated in Figure 2-48.

2.5.2 Validation And Release Criteria

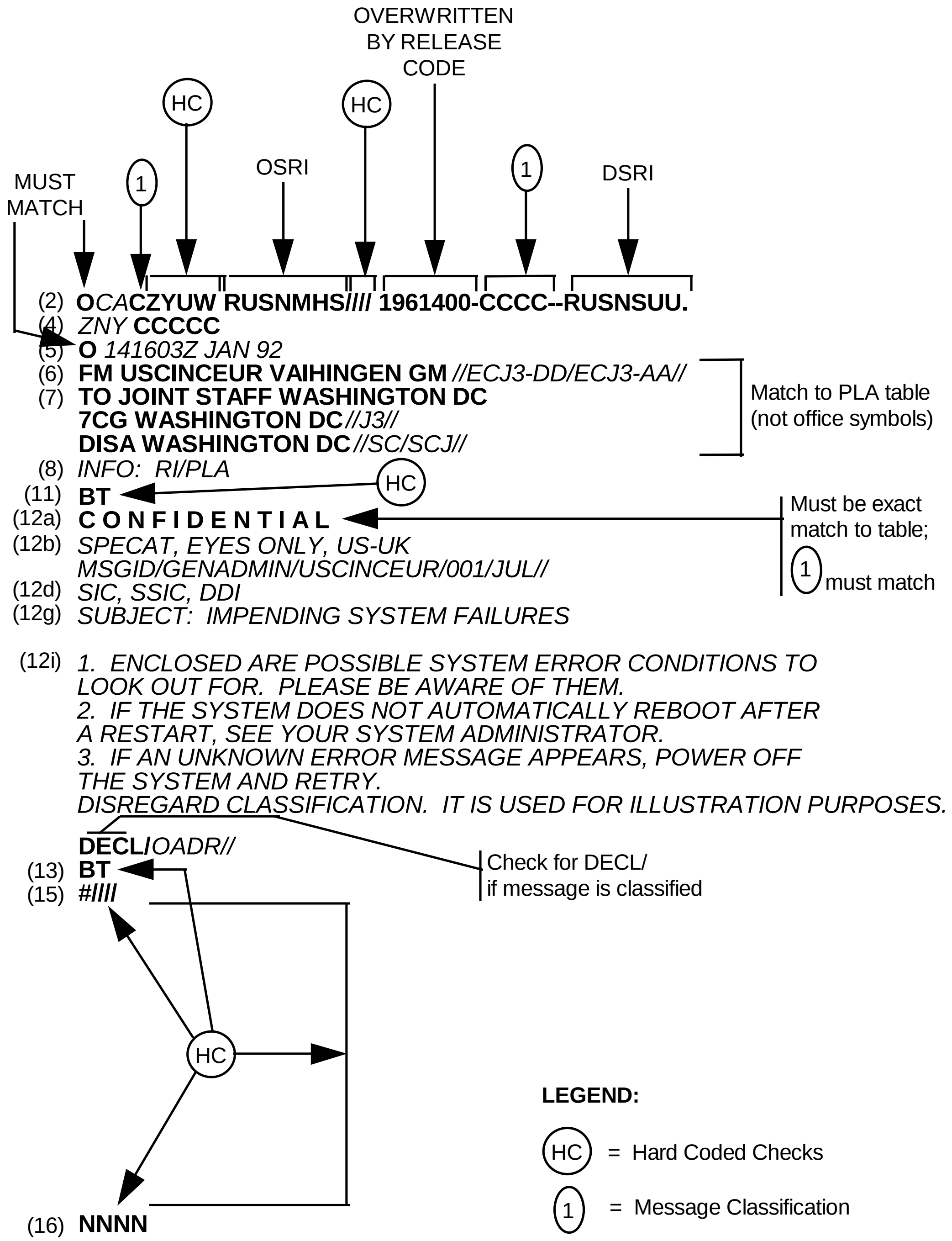
The validation of the message by MTF Save/Validate is basically the same process as the Applix Words Editor check button (✓) next to the Display Security button. Both the Words check button and the MTF Editor Save_Validate selections will pop up the Validation Errors window (Figure 2-46) and describe the errors by line number. After the message is drafted and pre-validated, it is routed for approval and release with a buckslip (Figure 2-44). The person with release authority will review the message and use the Message Manager validation tools to determine if the message meets the release criteria. Once this is done he will release the message to the SAT/CBT for transmission to the AUTODIN message delivery system. Message Manager validation uses the same validation criteria and the same configuration/data tables: Ri.CCA (Figure 2-50), Class.CCA (Figure 2-51), PLADATA.CCA, MAST_PLA.CCA (Figure 2-52) and PLAINDEX.CCA as well as non configurable checks consistent with the message format.

Figure 2-47 describes the format of a typical message with some detail about each component. Format lines 2 through 8 contain the routing and addressing information. Between format lines 11 and 13 is the body text of the message that begins with a plain language classification statement, UNCLAS, etc., and ends with declassification instruction if the message is classified. Format lines 15 and 16 end the message.

Figure 2-48 details each of the message components checked by the validation process. The only part of the body text that is checked is line 12a, for plain language classification information. This must be an exact character-by-character match, including spaces, with the table in the Class.CCA file. If the message is classified, the last line of the body text must include: DECL/.... (declassification instructions). A message will not validate if any of the tested criteria fails to match the criteria in the tables. The message will not validate if the Session Security Banner across the top of the desktop is less than the message classification in line 12a.

If there are site-specific Special Security Information Codes (SSICs) or similar special extensions needed on line 12a, the Class.CCA file must be edited to include these items (Example *CLASS=C,C O N F I D E N T I A L //N12234//),* which would validate line 12a as shown in Figure 2-48.

The message may fail the validation process for several reasons that do not necessarily make the message unreleasable or undeliverable. An example of this is when a drafter creates a message using a new PLA that has not been added to the AMHS or SAT PLA tables. This message can be released by the **Validate on Release = NO** option, and the System Administrator can add the PLA to the proper tables.


OVERWRITTEN BY RELEASE CODE
HC
HC
MUST MATCH
1
OSRI
1
DSRI
Match to PLA table (not office symbols)
HC
Must be exact match to table;
1 must match
Check for DECL/ if message is classified
HC
LEGEND:
HC = Hard Coded Checks
1 = Message Classification

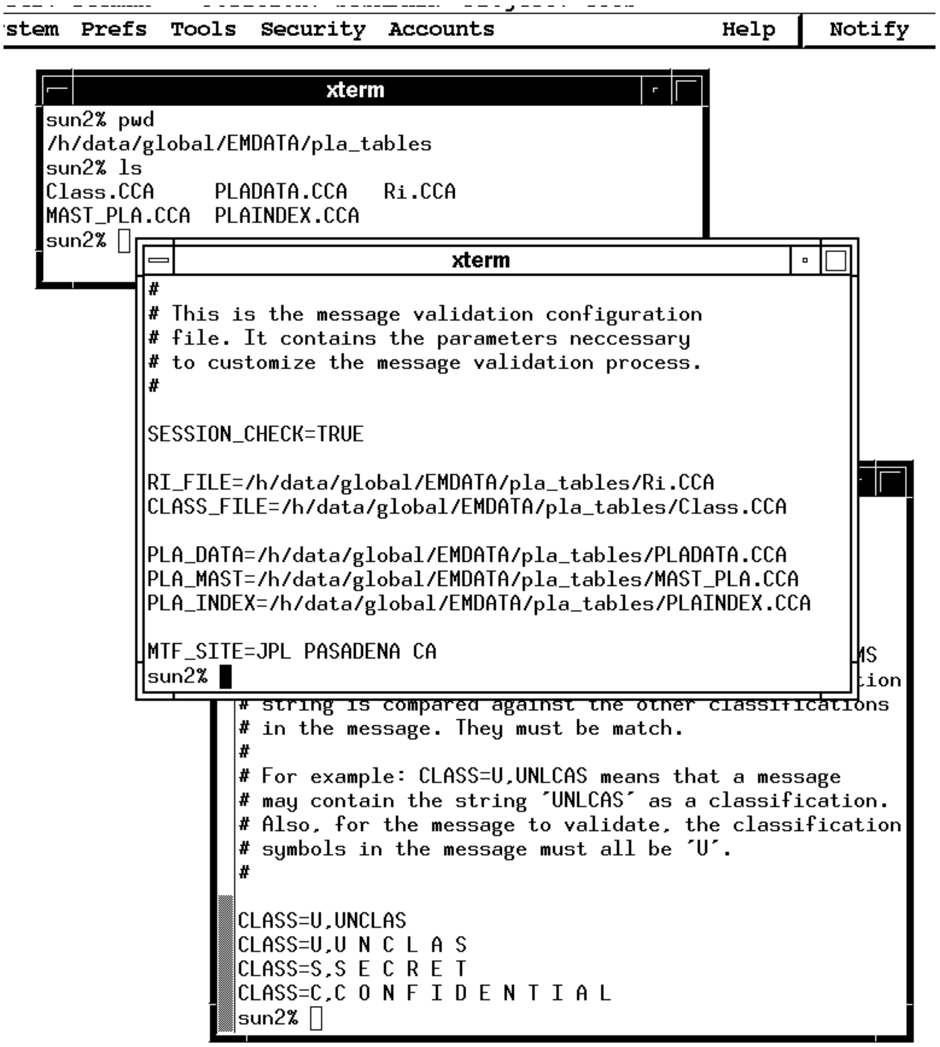
```
(2)   OCACZYUW RUSNMHS//// 1961400-CCCC--RUSNSUU.
(4)   ZNY CCCCC
(5)   O 141603Z JAN 92
(6)   FM USCINCEUR VAIHINGEN GM //ECJ3-DD/ECJ3-AA//
(7)   TO JOINT STAFF WASHINGTON DC
      7CG WASHINGTON DC//J3//
      DISA WASHINGTON DC//SC/SCJ//
(8)   INFO:  RI/PLA
(11)  BT
(12a) CONFIDENTIAL
(12b) SPECAT, EYES ONLY, US-UK
      MSGID/GENADMIN/USCINCEUR/001/JUL//
(12d) SIC, SSIC, DDI
(12g) SUBJECT:  IMPENDING SYSTEM FAILURES

(12i) 1.  ENCLOSED ARE POSSIBLE SYSTEM ERROR CONDITIONS TO
      LOOK OUT FOR.  PLEASE BE AWARE OF THEM.
      2.  IF THE SYSTEM DOES NOT AUTOMATICALLY REBOOT AFTER
      A RESTART, SEE YOUR SYSTEM ADMINISTRATOR.
      3.  IF AN UNKNOWN ERROR MESSAGE APPEARS, POWER OFF
      THE SYSTEM AND RETRY.
      DISREGARD CLASSIFICATION.  IT IS USED FOR ILLUSTRATION PURPOSES.
      DECL/OADR//
(13)  BT
(15)  #////

(16)  NNNN
```

Figure 2-48. Components of the Message Checked by Validation

2.5.3 Configuration Data Files (CDF) Maintenance

The Validation Criteria tables that are modifiable with an ASCII text editor (vi) are: Mv.CCA (Figure 2-49), Ri.CCA (Figure 2-50), Class.CCA (Figure 2-51), and MAST_PLA.CCA (Figure 2-52). As discussed in the previous section several other criteria are hard coded into the validation process. After editing MAST_PLA.CCA it is necessary to run the create_pla_files program to create an updated PLADATA.CCA and PLAINDEX.CCA. The SAT/CBT PLA tables should be updated to match MAST_PLA.CCA using the J:\AUTODIN\PLAEDIT program as described in Section 3 of the SAT/CBT System Users Manual.

The Master Validation Configuration Data file (CDF) located in the **/h/CCAPS/data/config/Mv.CCA** file defines the site-specific paths to the various validation data files. Figure 2-39 shows a typical CDF file. The CDF file is used to assign values to various token names used by the validation algorithms.



```
sun2% pwd
/h/data/global/EMDATA/pla_tables
sun2% ls
Class.CCA     PLADATA.CCA   Ri.CCA
MAST_PLA.CCA  PLAINDEX.CCA
sun2% 
```

```
#
# This is the message validation configuration
# file. It contains the parameters neccessary
# to customize the message validation process.
#

SESSION_CHECK=TRUE

RI_FILE=/h/data/global/EMDATA/pla_tables/Ri.CCA
CLASS_FILE=/h/data/global/EMDATA/pla_tables/Class.CCA

PLA_DATA=/h/data/global/EMDATA/pla_tables/PLADATA.CCA
PLA_MAST=/h/data/global/EMDATA/pla_tables/MAST_PLA.CCA
PLA_INDEX=/h/data/global/EMDATA/pla_tables/PLAINDEX.CCA

MTF_SITE=JPL PASADENA CA
sun2% 
```

```
                                                 MS
                                               tion
# string is compared against the other classifications
# in the message. They must be match.
#
# For example: CLASS=U,UNLCAS means that a message
# may contain the string 'UNLCAS' as a classification.
# Also, for the message to validate, the classification
# symbols in the message must all be 'U'.
#

CLASS=U,UNCLAS
CLASS=U,U N C L A S
CLASS=S,S E C R E T
CLASS=C,C O N F I D E N T I A L
sun2% 
```

Figure 2-49. Configuration Data File Mv.CCA

The SESSION_CHECK token is used to determine if the validation routines will validate the security markings of a message against the classification of the workstation. A value of TRUE will validate the message against the classification of the workstation. A value of FALSE will not validate the message against the workstation classification. The workstation security classification is displayed by the Session Security Banner across the top of the desktop and is set by the GCCS Security Manager in **/h/EM/Security/classification** file.

The RI_FILE token is used to specify the exact location of the Routing Indicator Configuration File (RICF). The Ri.CCA file (Figure 2-50) contains two variable values—the Originating Station Routing Indicator (OSRI) on line 1 and the Destination Station Routing Indicator (DSRI) on line 2. The OSRI is the routing indicator (RI) code for the site, and the DSRI is the RI for the AUTODIN switch connected to the SAT.

```
RUSNJPL
RUSNASA
#
# This file contains the first and second routing
# indicators that are used in validating the first
# line of a release message.
#
# The routing indicators must appear in the first
# two lines of this file.
#
```

Figure 2-50. Routing Indicator Configuration File Ri.CCA File

The CLASS_FILE token is used to specify the exact location of the Classification Configuration File (CCF). The Class.CCA file (Figure 2-51) contains the variable value CLASS_SYMS=X,X,X, X, X. These are alphanumeric pairs made up of the class value letter and the Monitor & Control (MAC) security value number. There can be more alphanumeric pairs than classes defined. The check for DECL on line 12i is triggered by a 2 through 4 in this entry.

```
# This file contains the valid classifications
# for the message validation routines. Each entry
# contains the
#
# The CLASS_SYMS variable contains all the valid
# classification symbols along with the value
# of each symbol. Each symbol listed represents
# a valid symbol that can appear in the message (see
# description of format line 2 for an example). The
# numeric value of symbol is used to make sure the
# falls within the security of monitor and control (MAC).
# If the SESSION_CHECK option is enabled, the numeric
# values are compared against the monitor and control MAC)
# security values.
# The MAC security values are as follows:
#
#        Top Secret       = 4
#        Secret           = 3
#        Confidential     = 2
#        Unclassified     = 1
#        No Classification = 0
#
# For example: U1 means that the symbol 'U' may appear
# as a classification symbol with a message. The '1'
# means that the symbol will be treated as Unclassified
# when compared against the MAC security classification.
#
# Symbols are always alphanumeric and values are
# always numeric.

CLASS_SYMS=E1,U1,C2,S3

# The CLASS variables list the valid classification
# strings that may appear within the message. Along
# with each string is a classification symbol. The
# symbol must be one of the symbols on the CLASS_SYMS
# line. When a message is validated, the classification
# string is compared against the other classifications
# in the message. They must be match.
#
# For example: CLASS=U,UNLCAS means that a message
# may contain the string 'UNLCAS' as a classification.
# Also, for the message to validate, the classification
# symbols in the message must all be 'U'.

CLASS=U,UNCLAS
CLASS=U,U N C L A S
CLASS=S,S E C R E T
CLASS=C,C O N F I D E N T I A L   (SSIC Codes could be added here)
```

Figure 2-51. Classification Configuration File Class.CCA File

The MAST_PLA.CCA file (Figure 2-52) is a plain ASCII text file with information supplied to the site by the Communication Group. Update notices need to be edited into the file with a common ASCII text editor such as vi. Changes made to this file must also be made to the SAT/CBT PLA tables using the PLAEDIT program as described in the SAT/CBT manual. After the editing is complete, the create_pla_files utility located in **/h/CCAPPS/progs** must be run to update the PLADATA.CCA binary data file and PLAINDEX.CCA hash table for use by the validation process. For the changes to take effect, the users must exit MTF Editor and Message Manager and restart their applications. In most cases this will occur when the user logs out at the end of shift.

```
1 ACCS OFFUTT AFB NE                115 TASG TRUAX FLD MADISON WI
1 ACOMMG OFFUTT AFB NE              116 TFW DOBBINS AFB GA
1 BEAWFWG BEAUVECHAIN               117 TRW BIRMINGHAM MUNI APRT AL
1 CSG LANGLEY AFB VA                119 FIG HECTOR FLD FARGO ND
1 GEAIRDIV MESSTETTEN               12 ABG RANDOLPH AFB TX
1 GESSMWG LANDSBERG                 12 AF BERGSTROM AFB TX
1 ISG LINDSEY AS GM                 AIG 862
1 RCHA LAHR GM                      AIG 8722
1 SPACE SPT GP PETERSON AFB CO      AIG 8723
1 STROC                             AIG 8728
1 TFW LANGLEY AFB VA                AMCONSUL CHIANG MAI
10 AF BERGSTROM AFB TX              AMCONSUL CURACAO
10 TRW RAF ALCONBURY UK             AMCONSUL DHAHRAN
10 TRWCP RAF ALCONBURY UK           AMEMBASSY BUCHAREST
100 AREFW BEALE AFB CA              AMEMBASSY BUDAPEST
100 CSG BEALE AFB CA                HQ ESS OFFUTT AFB NE
102 FIW OTIS ANG BASE MA            HQ EST LANGLEY AFB VA
103 TFG BRADLEY ANGB CT             JDSCC WASH DC
107 FIG NIAGARA FALLS NY            JDSCC WASHINGTON DC
108 TFW MCGUIRE AFB NJ              NAVPTO WASH DC
11 SGCP RAF FAIRFORD UK             NAVPTO WASHINGTON DC
1100 ABG BOLLING AFB DC             RS BREKENDORF GM
113 TFW ANDREWS AFB MD              USACIDC WASHINGTON DC
114 TFG JOE FOSS FLD SD             USS IWO JIMA
1141 USAFSAS VAIHINGEN GM           ZAMISH KINSHASA CG
```

Figure 2-52. Sample of MAST_PLA.CCA File

The MTF_SITE token (Figure 2-49) is used to specify the Plain Language Address of the local AMHS site. The value of this parameter is used as the default FM PLA when creating new messages in the MTF Editor. The list is only a representative sample of typical PLAs. You should get a complete file from DISA.

When editing of the message validation configuration files is complete, the AMHS System Administrator should have his GCCS System Administrator notify all logged-in users to exit and restart their Message Manager, MTF Editor and Applix Words applications. Changes to the configuration files do not take effect until the applications that use them are restarted. The System Alarm feature of the System Monitor can be used for status/notification messages, or this could be done with the E-mail system.

To determine who is logged in, use the current user pull-down from EM System Monitor (reference Appendix C, Section C.3.5) and generate a System Alarm explaining to the users that if they are using Message Manager or MTF Editor, they need to exit and restart those programs to update their PLA table access.

The authority to release messages to AUTODIN is granted on an individual basis (in other words, by userid). Release authority is controlled using UNIX group permissions and is tied to a UNIX group called 'amh_rel'. Any user that is a member of this group has the authority to release messages. Since access to UNIX groups is managed by GCCS System Administrators, AMHS System Administrators should work with their GCCS System Administrators to ensure that the proper users have authority to release messages.

2.6 COMEBACK COPY

2.6.1 Comeback Copy Process

Although a comeback copy is generated from an outbound message, it is considered an inbound message because it is put into the message database by the **cbc_feed** process as well as profiled to the message drafter and any coordination points. In Figure 2-53, when a message is released by the Message Manager Release software to the SAT/CBT, a separate file called the coordination record (or message action record (MAR)) is created with the message drafter ID, the coordination points and comments, the message release authority (MRA) name, and the time stamp of when the message was released. This coordination record/MAR file is put into directory **/h/AMHS/Server/topic/amhs_db/coord/comeback/**. The **cbc_feed** looks for coordination records/MAR files in this directory. Once a MAR is available the **cbc_feed** looks for the corresponding file name in the SAT/CBT transmit archive directory. It then copies the message into the message database and sends a notification to the TOPIC database builder process for the **cbc_feed** (**dp4**). The **cbc_feed** also has its own TOPIC partition merger called **mg4**.

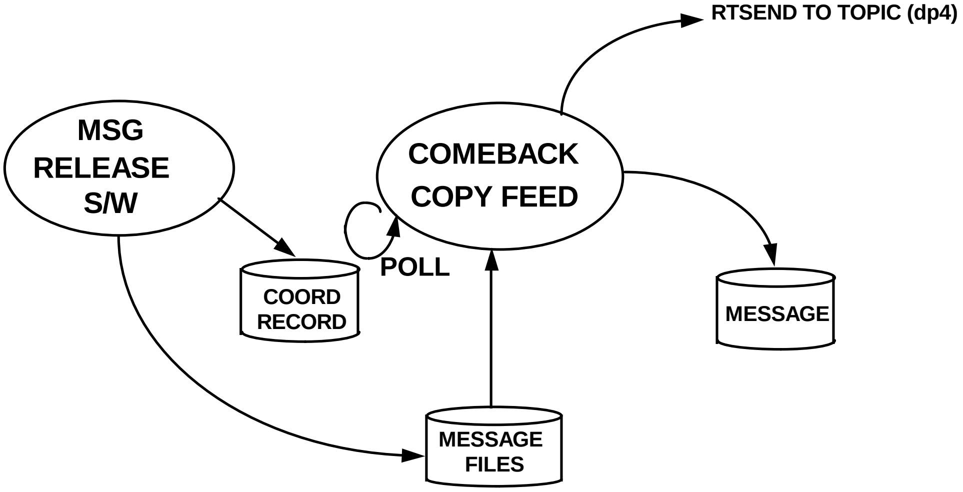


Figure 2-53. Coordination Record Process

2.6.2 Comeback Copy Feed Data Flow

Transmitted AUTODIN messages are introduced back into the AMHS by a process called the Comeback Copy Feed (**cbc_feed**), as shown in Figure 2-54 and described below. As messages are released by the Message Manager software, they are stored in the **/h/AMHS/Server/sat/autodin/xmit** directory. In addition, the coordination record which is generated when the drafter composes the message is stored in the **/h/AMHS/Server/topic/amhs_db/coord/comeback** directory. The Comeback Copy Feed software checks the received coordination record files against the messages in the SAT/CBT transmit archive. When a message appears in the SAT/CBT transmit archive, it means that a positive acknowledgment has been received for the message from the AUTODIN interface. The **cbc_feed** appends the coordination record to the message and feeds the message into TOPIC, where a profile delivers the message to the **DRAFTER, COORDINATION POINTS** and **RELEASER** of the message.

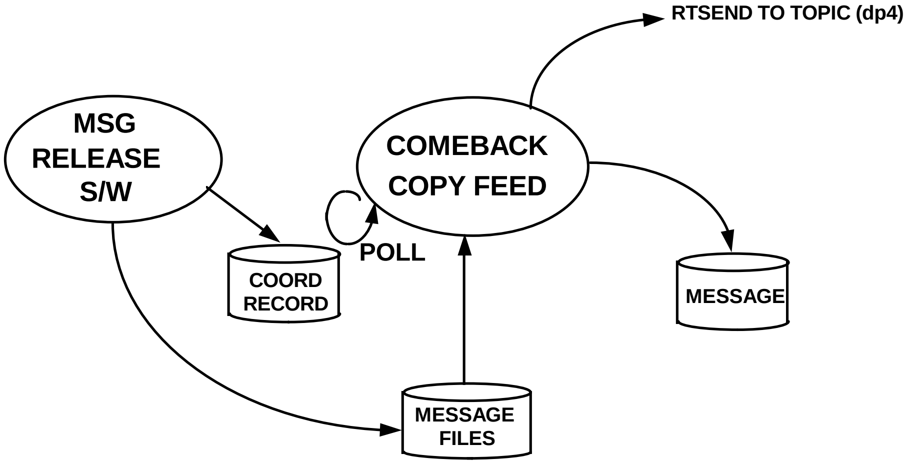


Figure 2-54. Comeback Copy Feed Data Flow

2.7 MESSAGE NOTIFICATION

When outgoing messages are released successfully, a **Released OK** window is opened to notify the releaser. On incoming and CBC messages the **Notify** and **Alarm** indicators on the GCCS Desktop tool bar can be configured to respond to different types as they appear in the user's message queues. **INFO** messages would typically trigger the **Notify** indicator and **ACTION** messages would flash the **Alarm** indicator **Red**. The user with AMHS needs would normally leave a small message browser window open in the corner of their screen. The message browser window updates automatically in real-time.

2.8 AMHS FILE SYSTEM

As discussed in Section 2.2.1, the underlying strength of the UNIX environment is the inherent security, flexibility and power of the file system. Each directory, executable program and data element is accessible only if appropriate permissions exist.

Permission Field	# of Links	File's Owner	File's Group	Size in Bytes	Date of Last Modification	Filename
drwxrwxrwx	4	E6JSF	staff	1024	Nov 12 12:03	Manpower

The Permissions field is 10 char; 1) -/d=file/directory, 234) user, 567) group, 8910) others, r=read,w=write,x=exec

The AMHS makes use of this for many system security features. The AMHS is able to run the standard COTS SAT/CBT software without any modification by NFS mounting the SAT/CBT local hard drive on the AMHS Server and running a process within the **sat_feed** looking for files being written to its hard drive. When the process sees a new file in the AUTODIN subdirectory it creates a token and hands the token to the **sat_feed** for processing. This keeps the SAT/CBT DMS certification intact.

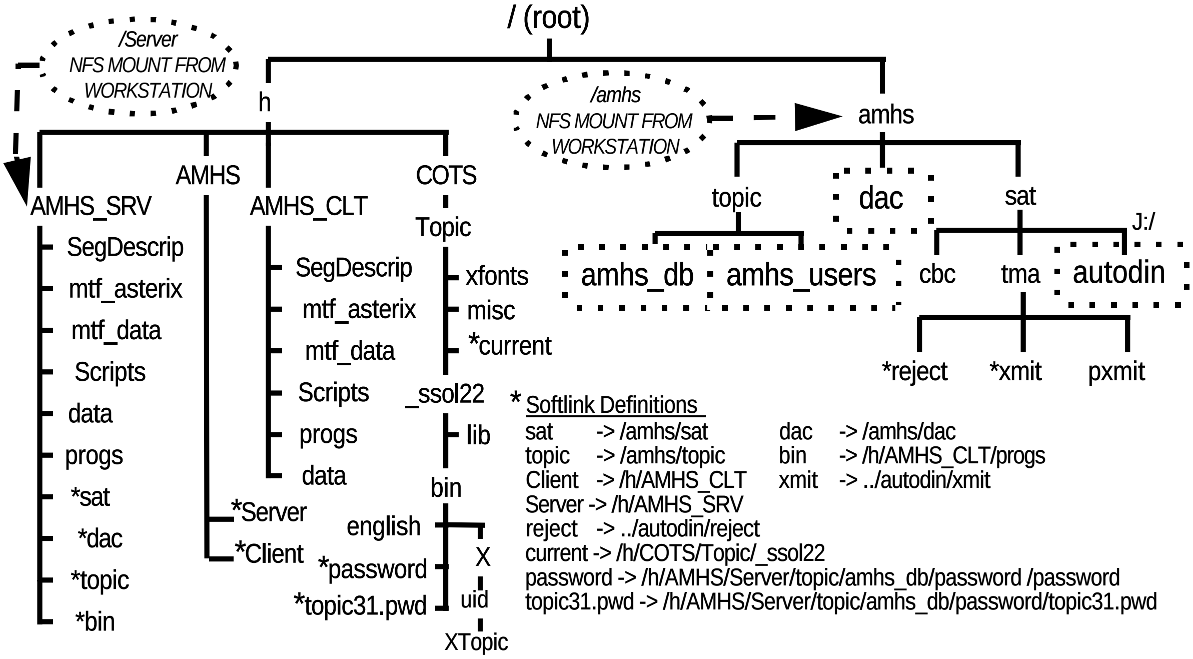
To assist in understanding the paths shown with the executables and files described throughout this document Figures 2-55 and 2-56 are included. This is not a complete top down listing, but rather an overview of all of the higher levels of the AMHS file system.

2.8.1 Server File Directory Tree

The number of directories and paths in the server file system requires the tree be shown in multiple diagrams. The first diagram depicts the top levels and subsequent diagrams show more detail for four of the paths. These five diagrams form the AMHS Server Directory Structure.

Figure 2-55 AMHS Server Directory Structure

Figure 2-55a	/(root) directory
Figure 2-55c	/dac path
Figure 2-55d	/amhs_db path
Figure 2-55d	/amhs_users path
Figure 2-55e	/autodin path



* Softlink Definitions

sat -> /amhs/sat
topic -> /amhs/topic
Client -> /h/AMHS_CLT
Server -> /h/AMHS_SRV
reject -> ../autodin/reject
current -> /h/COTS/Topic/_ssol22
password -> /h/AMHS/Server/topic/amhs_db/password /password
topic31.pwd -> /h/AMHS/Server/topic/amhs_db/password/topic31.pwd
dac -> /amhs/dac
bin -> /h/AMHS_CLT/progs
xmit -> ../autodin/xmit

Figure 2-55a. /(root) directory

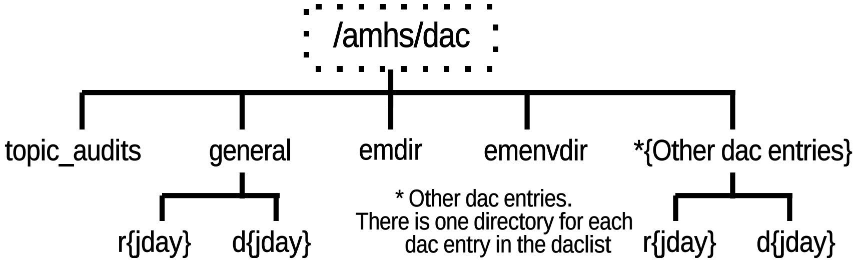


Figure 2-55b. dac path

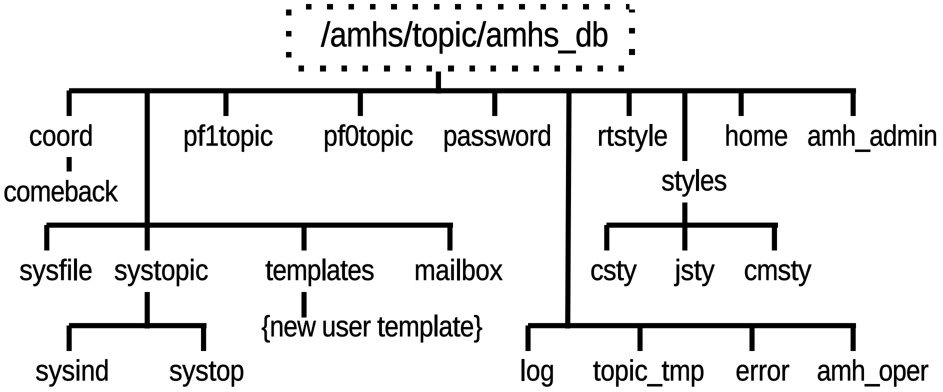


Figure 2-55c. amhs_db path

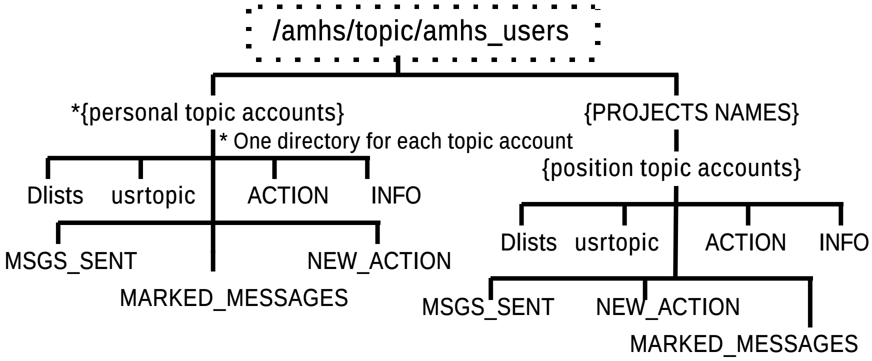


Figure 2-55d. amhs _users path

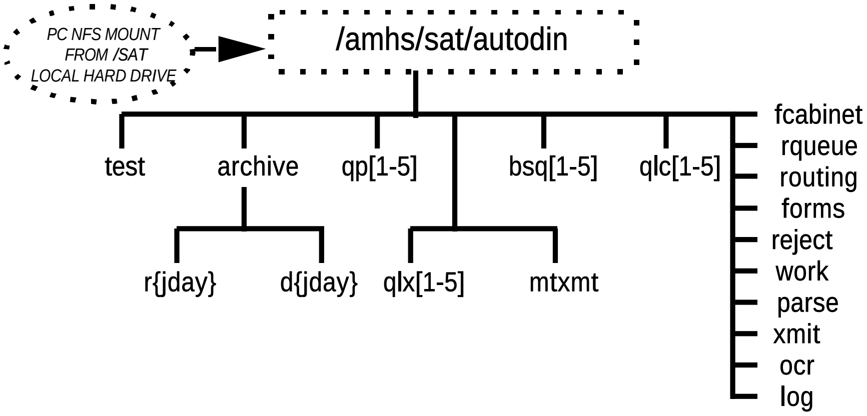
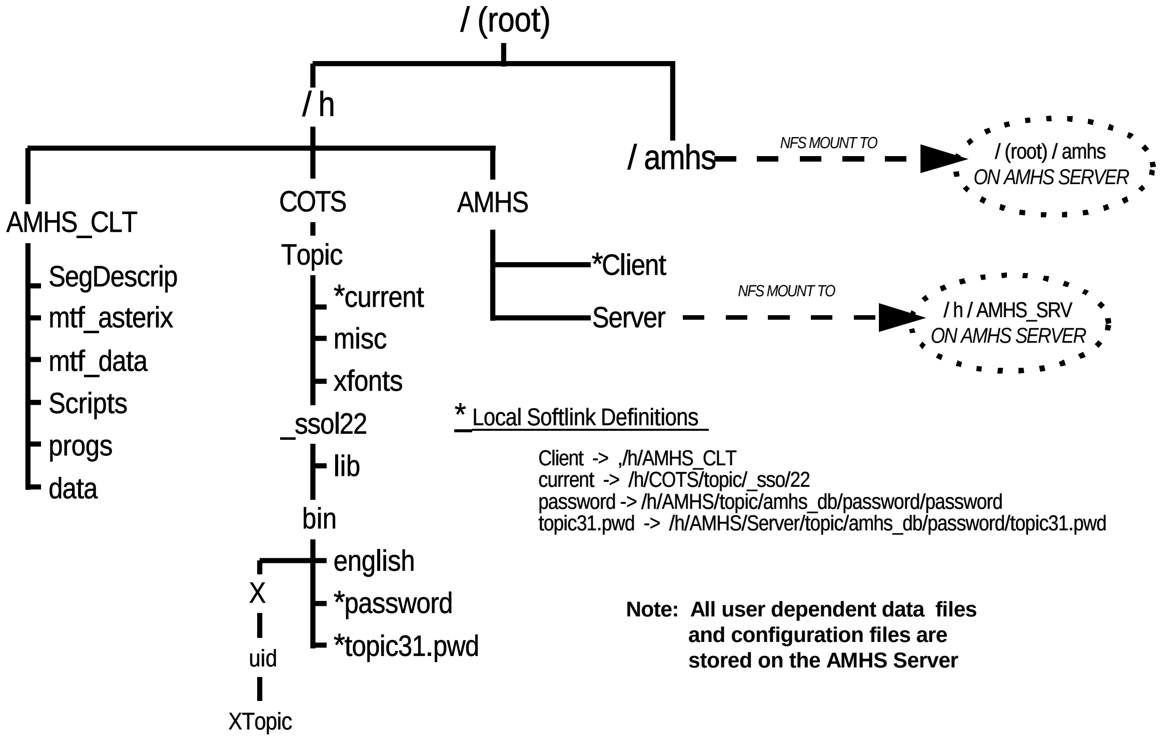


Figure 2-55e. /autodin

2.8.2 User Workstation File Structure

The NFS mounts on the client workstation make it possible for each user's configuration and preferences to be stored on the AMHS Server. This feature allows a user to log on to any machine that is the same basic type of hardware and run with the same look and feel as their home workstation. See Figure 2-56.



* Local Softlink Definitions

Client -> ,/h/AMHS_CLT
current -> /h/COTS/topic/_sso/22
password -> /h/AMHS/topic/amhs_db/password/password
topic31.pwd -> /h/AMHS/Server/topic/amhs_db/password/topic31.pwd

Note: All user dependent data files and configuration files are stored on the AMHS Server

Figure 2-56. AMHS User Workstation/Client Directory Structure

2.9 SECURITY - AUDIT

Requirements for auditing the GCCS AMHS is governed by the Defense Message System (DMS) of Defense Information Systems Agency (DISA). All DMS security requirements are covered in the GCCS AMHS DMS Component Approval Process (CAP) documentation that was used to certify this system. The GCCS AMHS Trusted Facility Manual (TFM) has guidance for security issues as will the site System Security Officer (SSO) for local policy and procedures.

Audit logs must be protected from modification and unauthorized access or destruction. There are mandatory audit events which include: success and failure of log on and log off, any violations of security policy, and release of messages. Events must also be audited such as the date and time, user, type of event, address or origin of request, and the success or failure of the event. Moreover, the System Administrator must be able to selectively audit the actions of one or more users based on individual identity. Additionally, the audit trail must have sufficient detail to reconstruct events in determining the cause and magnitude of compromise should a security violation occur. Finally, audit files must be readily accessible for a minimum of 30 days and must be archived for a minimum of one year.

There are several audit logs created by the suite of AMHS software that will allow the System Administrator/SSO to sufficiently trace the activities of users as well as the history of a message as it moves through the AMHS. Audit logs are created by the COTS product “TOPIC” and the backside to AUTODIN—SAT/ CBT. The AMHS **sat_feed** and **cbc_feed** also create audit logs which contain information about every message that is processed. There are also operating system level audit logs which are created by the Solaris 2.3 Basic Security Module (BSM) audit daemon which runs on each GCCS Server and each GCCS workstation that has access to the AMHS. The AMHS provides an audit reporting feature of the Solaris BSM collected audit, which is available in the Security Manager software from the Executive Manager (EM) segment. Figure 2-57 depicts the messaging audit logs, their physical location and file name. Figure 2-58 depicts the user access auditing logs, their physical location and the file name.

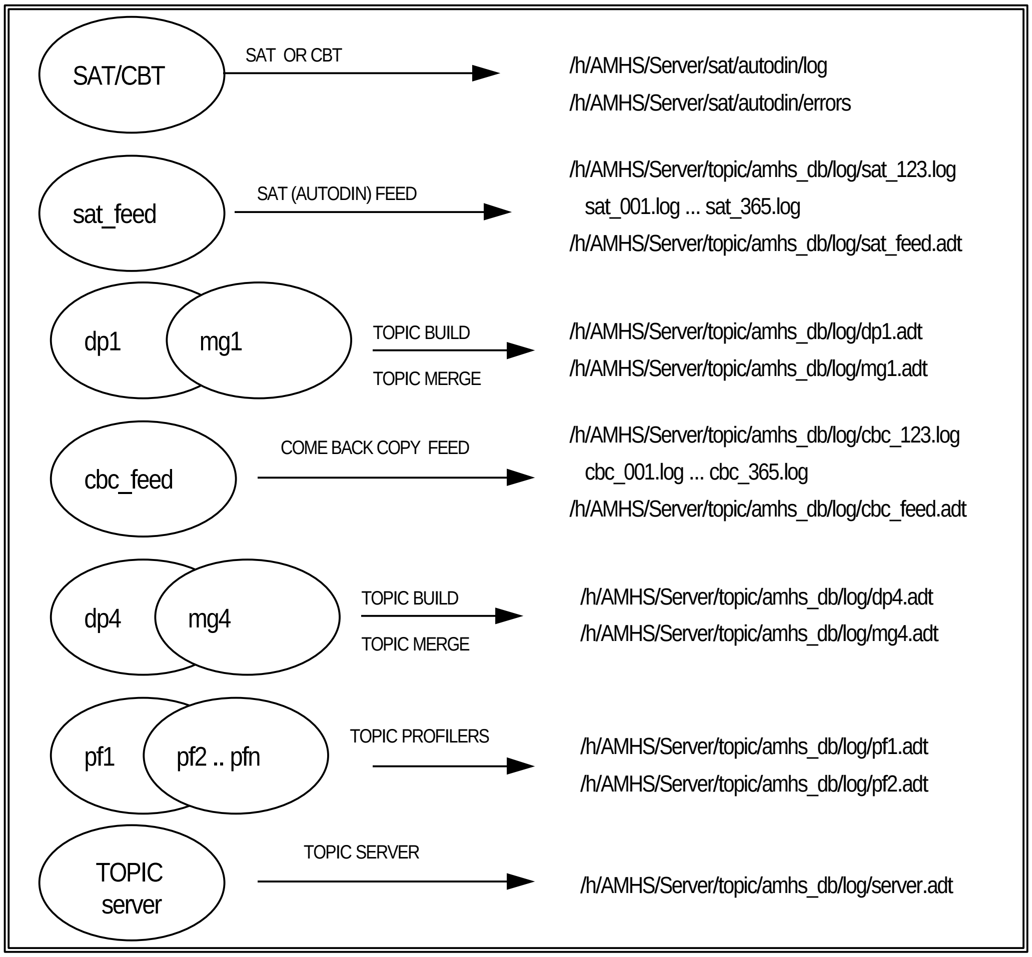

SAT/CBT
SAT OR CBT
/h/AMHS/Server/sat/autodin/log
/h/AMHS/Server/sat/autodin/errors
sat_feed
SAT (AUTODIN) FEED
/h/AMHS/Server/topic/amhs_db/log/sat_123.log
sat_001.log ... sat_365.log
/h/AMHS/Server/topic/amhs_db/log/sat_feed.adt
dp1
mg1
TOPIC BUILD
TOPIC MERGE
/h/AMHS/Server/topic/amhs_db/log/dp1.adt
/h/AMHS/Server/topic/amhs_db/log/mg1.adt
cbc_feed
COME BACK COPY FEED
/h/AMHS/Server/topic/amhs_db/log/cbc_123.log
cbc_001.log ... cbc_365.log
/h/AMHS/Server/topic/amhs_db/log/cbc_feed.adt
dp4
mg4
TOPIC BUILD
TOPIC MERGE
/h/AMHS/Server/topic/amhs_db/log/dp4.adt
/h/AMHS/Server/topic/amhs_db/log/mg4.adt
pf1
pf2 .. pfn
TOPIC PROFILERS
/h/AMHS/Server/topic/amhs_db/log/pf1.adt
/h/AMHS/Server/topic/amhs_db/log/pf2.adt
TOPIC server
TOPIC SERVER
/h/AMHS/Server/topic/amhs_db/log/server.adt

Figure 2-57. Message Activity Auditing Logs And Location

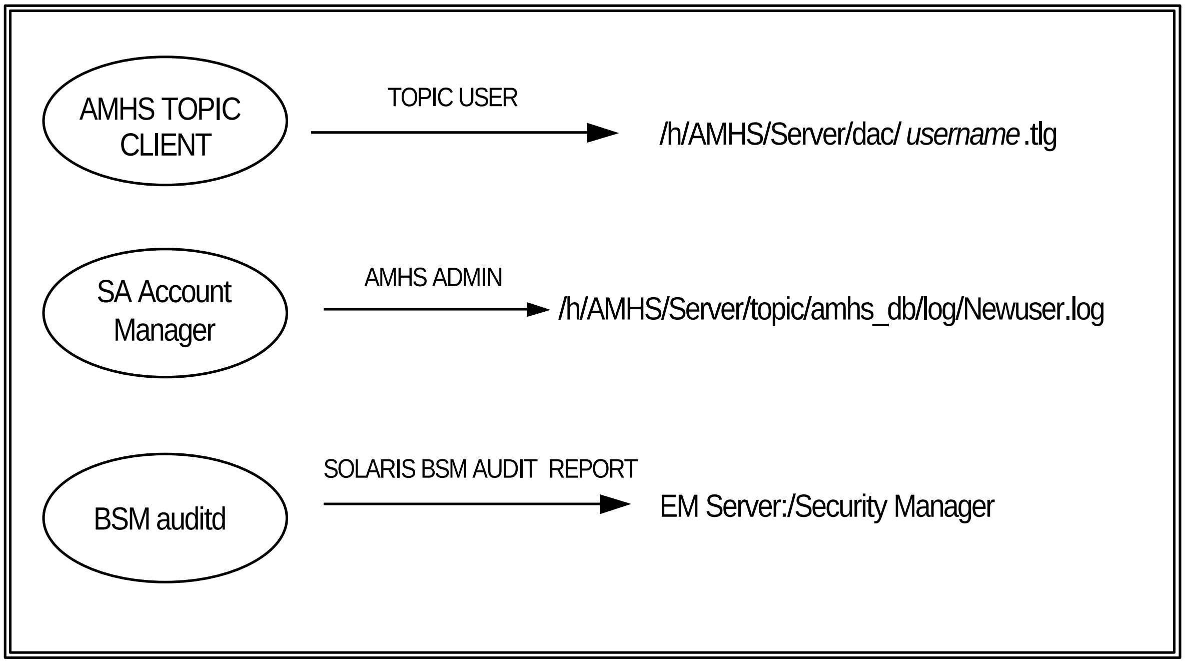

AMHS TOPIC CLIENT
TOPIC USER
/h/AMHS/Server/dac/ *username* .tlg
SA Account Manager
AMHS ADMIN
/h/AMHS/Server/topic/amhs_db/log/Newuser.log
BSM auditd
SOLARIS BSM AUDIT REPORT
EM Server:/Security Manager

Figure 2-58. User Access Auditing Logs And Location

Typically, the Security Officer will review the audit logs, but it is the System Administrator who will use the logs to check the movement of messages or review the logs for error conditions as well as maintain the required 30 days on-line of auditing. This discussion will cover each type of audit log within the AMHS and how the System Administrator or System Security Officer might use them. Proof of meeting the DMS auditing requirements was demonstrated during the formal DMS Certification testing for the GCCS AMHS.

As a rule all audit logs should be checked daily for ERROR conditions. This can be done using the UNIX “grep” utility to search for “ERROR”. Another very important use of logs in a real-time messaging environment is to quickly identify why processes may continually die and what the solution might be.

2.10 AUTODIN MESSAGE AUDITING

2.10.1 SAT/CBT Audit Logs

When a message is received at the SAT/CBT an entry is made in the SAT/CBT log. The information includes the Julian date, the time stamp, receive (R) vs. transmit (T), FL2, and the location of the message in the SAT/CBT archive. Figure 2-59 shows an example of the SAT/CBT audit log for both transmitted and received AUTODIN messages. The SAT/CBT also logs error conditions as well as logons and the operator ID (does not audit logoff). The log can be printed on the SAT/CBT printer by selecting the print log feature for either daily log or current log from the SAT/CBT menuing system.

The System Administrator will most likely print a daily report, check for any rejected messages, and store it in a binder for reference. A user might also request the time a message was released, but this information is also available in the comeback copy of the message that should have been delivered to the drafter.

NOTE: As a standard operating procedure the System Administrator should check the SAT/CBT error queue status several times daily to correct/process the messages into the AMHS and not rely on checking the log report at the close of and/or beginning of the work day.

```
123 16:41 R*** RCAUZYUW RHFQAAA0016 1231633-UUUU--RHFQAAA.
                               ARCHIVE\R123\261506.035
123 16:46 R*** OCAUZYUW RHFQAAA0017 1231633-UUUU--RHFQAAA.
                               ARCHIVE\R123\261507.011
123 16:52 R*** RCAUZYUW RHFQAAA0018 1231633-UUUU--RHFQAAA.
                               ARCHIVE\R123\261508.000
123 16:52 R*** RCAUZYUW RHFQAAA0016 1231633-UUUU--RHFQAAA.
                               ARCHIVE\R123\261508.001
123 16:54 T*** ZCAUZYUW RHFQAAA0016 1231633-MTMS-UUUU--RHFQAAA.
                               ARCHIVE\T123\D16435.MSG
123 16:55 R*** RCAUZYUW RHFQAAA0016 1231633-UUUU--RHFQAAA.
                               ARCHIVE\R123\261509.002
```

Figure 2-59. SAT/CBT Audit Log

2.10.2 SAT_FEED Audit Logs

If the **sat_feed** is running (and it should always be running) **sat_feed** logs are generated daily and named with the Julian date, e.g. sat_001.log, sat_002.log ... sat_365.log. The logs contain quite a bit of information about an incoming message. Referring to Figure 2-60, the version of the **sat_feed** is indicated when it is first started. It will tell you the time stamp of when the **sat_feed** began processing the message (when the message token was read in the SAT/CBT backside queue), the location of the message in the SAT/CBT archive, DTG, classification, precedence, security information from FL2 and FL4, the discretionary access control (DAC) mask that is applied, where the message was put in the message database, and the notification to TOPIC (**dp1**) that a new message is available. If there are any information parsing problems for the **sat_feed** or if it suspects a duplicate message, the error condition is written in the audit log and the message is moved to a **sat_feed** error directory in the AMHS Real-Time directory. Figure 2-60 shows an example of a **sat_feed** audit log. The audit log can display information about sectioned messages as well as errors about how sections are received. Figure 2-60 also shows a portion of the log for a sectioned message. The **sat_feed** waits on all of the sections to arrive before it sends the message into the TOPIC database, unless a time-out value has been exceeded (site-configurable in **vardef**).

These are useful logs and will be consulted often. The System Administrator will reference them to trace a message into the system but should also check daily for any ERROR entries. If an error condition has occurred, the audit log will indicate that the message was moved to the **sat_feed** error directory in the AMHS Real-Time directory and what caused the error (the directory is located: **/h/AMHS/Server/topic/amhs_db/error/sat_feed**).

NOTE: The System Administrator should check the **h/AMHS/Server/topic/amhs_db/error/sat_feed** directory daily to make corrections where needed and feed the message into the database.

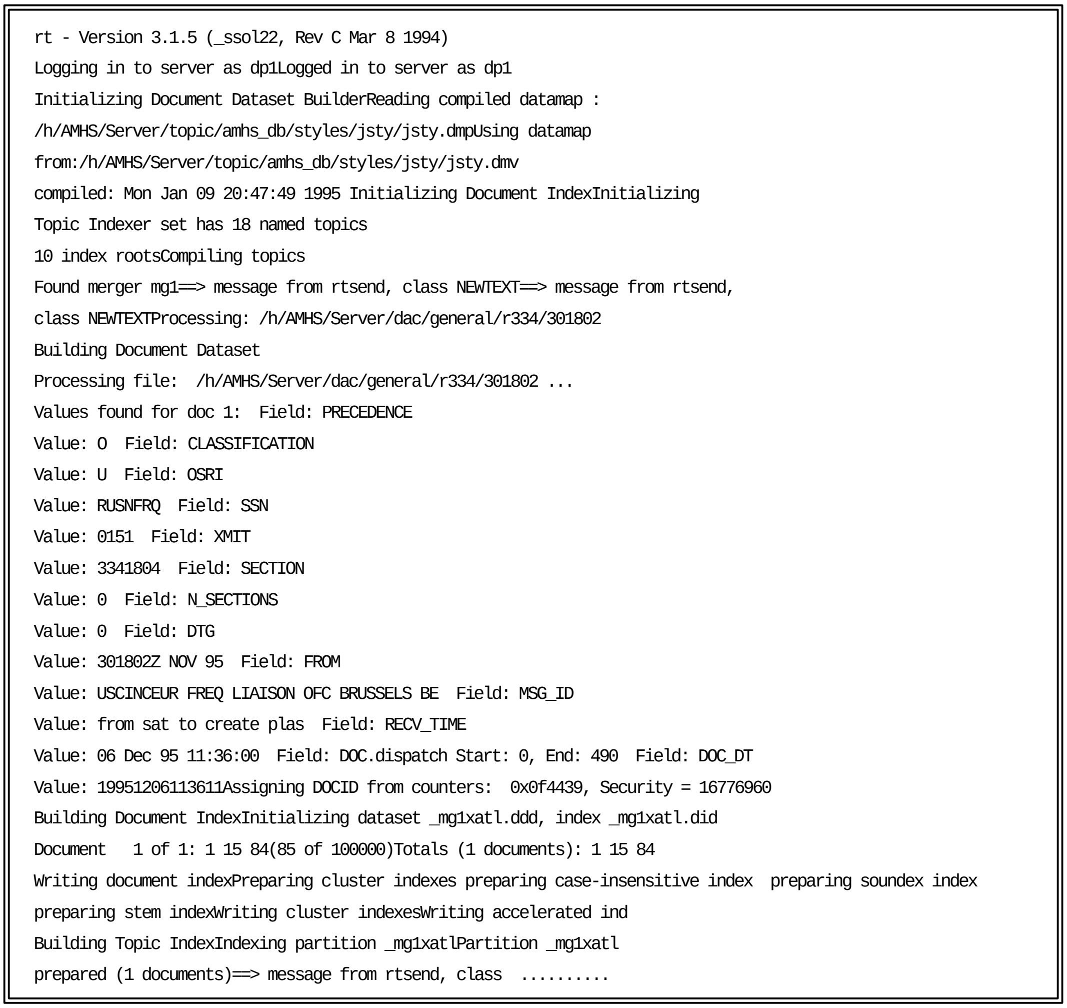
```
Dec 6 11:18:23 INFO:  sat_feed started  Version: 3.68

Dec 6 11:36:11 INFO:  sat_feed: Message token: /h/AMHS/Server/sat/autodin/bsq3/00002816
Dec 6 11:36:11 INFO:  sat_feed:  Received: /h/AMHS/Server/sat/autodin/archive/r334/301802
Date/Time Group: 301802Z NOV 95  Precedence:  O  Class: U
Format line 2:    OAAUZYUW RUSNFRQ0151 3341804-UUUU--RUSNSOW RUSNTRW.
Format line 6:    FM USCINCEUR FREQ LIAISON OFC BRUSSELS BE
Access type: General  TOPIC Mask:  0x00ffff00
Dec 6 11:36:12 INFO: sat_feed: Sent to Topic: /h/AMHS/Server/dac/general/r334/301802

Dec 6 11:36:12 INFO:  sat_feed: Message token: /h/AMHS/Server/sat/autodin/bsq3/00010525
Dec 6 11:36:12 INFO:  sat_feed:  Received: /h/AMHS/Server/sat/autodin/archive/r334/152245
Date/Time Group: 152245Z JUN 95  Precedence:  O  Class: U
Format line 2:    OAAUZYUW RUSNFRQ0152 3341808-UUUU--RUSNSOW RUSNTRW.
Format line 6:    FM USCINCEUR FREQ LIAISON OFC BRUSSELS BE
Access type: Limdis  TOPIC Mask:  0x00001000
Dec 6 11:36:12 INFO:  sat_feed:   Received: /h/AMHS/Server/sat/autodin/archive/r334/152245
ADDING NEW FAMILY MESSAGE: 152245Z JUN 95 RUSNFRQ U 6  SECTION 1 OF 6    OSSN: 0152

Dec 6 11:36:12 INFO:  sat_feed: Message token: /h/AMHS/Server/sat/autodin/bsq3/00012548
Dec 6 11:36:13 INFO:  sat_feed:  Received: /h/AMHS/Server/sat/autodin/archive/r334/152245.000
Date/Time Group: 152245Z JUN 95  Precedence:  O  Class: U
Format line 2:    OAAUZYUW RUSNFRQ0153 3341808-UUUU--RUSNSOW RUSNTRW.
Format line 6:    FM USCINCEUR FREQ LIAISON OFC BRUSSELS BE
Access type: General  TOPIC Mask:  0x00ffff00
Dec 6 11:36:13 INFO:  sat_feed: Sectioned Message
Received: /h/AMHS/Server/sat/autodin/archive/r334/152245.000
UPDATING FAMILY MESSAGE: 152245Z JUN 95 RUSNFRQ U 6   SECTION 2 OF 6    OSSN: 0153
```

Figure 2-60. sat_feed Audit Log

2.10.3 TOPIC Database Builder (dp1) Audit Log

The **sat_feed** has a TOPIC Database Builder process called **dp1** that extracts field information for each message, and assigns it a TOPIC message ID and writes it to its own audit log called **dp1.adt**. The **dp1** process also builds the real-time message partition and makes it available to users and the TOPIC Central Profiler. The **dp1** audit log is created, if not previously existing, when the process is first brought up and will continue to write to the same file until it is deleted and/or purged. Figure 2-61 has a sample of the audit information collected in the **dp1.adt** audit log file. This file can get very large so the System Administrator should watch the file size.

Although full of TOPIC information, the System Administrator will probably not find this log particularity interesting. Reference this log to explain why field information did not get properly stored for a message (subject line is a typical case) or in what partition a message got stored. It would also be used for checking performance during peak messaging periods for how many messages are getting stuffed into a partition.



```
rt - Version 3.1.5 (_ssol22, Rev C Mar 8 1994)
Logging in to server as dp1Logged in to server as dp1
Initializing Document Dataset BuilderReading compiled datamap :
/h/AMHS/Server/topic/amhs_db/styles/jsty/jsty.dmpUsing datamap
from:/h/AMHS/Server/topic/amhs_db/styles/jsty/jsty.dmv
compiled: Mon Jan 09 20:47:49 1995 Initializing Document IndexInitializing
Topic Indexer set has 18 named topics
10 index rootsCompiling topics
Found merger mg1==> message from rtsend, class NEWTEXT==> message from rtsend,
class NEWTEXTProcessing: /h/AMHS/Server/dac/general/r334/301802
Building Document Dataset
Processing file:  /h/AMHS/Server/dac/general/r334/301802 ...
Values found for doc 1:  Field: PRECEDENCE
Value: O  Field: CLASSIFICATION
Value: U  Field: OSRI
Value: RUSNFRQ  Field: SSN
Value: 0151  Field: XMIT
Value: 3341804  Field: SECTION
Value: 0  Field: N_SECTIONS
Value: 0  Field: DTG
Value: 301802Z NOV 95  Field: FROM
Value: USCINCEUR FREQ LIAISON OFC BRUSSELS BE  Field: MSG_ID
Value: from sat to create plas  Field: RECV_TIME
Value: 06 Dec 95 11:36:00  Field: DOC.dispatch Start: 0, End: 490  Field: DOC_DT
Value: 19951206113611Assigning DOCID from counters:  0x0f4439, Security = 16776960
Building Document IndexInitializing dataset _mg1xatl.ddd, index _mg1xatl.did
Document   1 of 1: 1 15 84(85 of 100000)Totals (1 documents): 1 15 84
Writing document indexPreparing cluster indexes preparing case-insensitive index  preparing soundex index
preparing stem indexWriting cluster indexesWriting accelerated ind
Building Topic IndexIndexing partition _mg1xatlPartition _mg1xatl
prepared (1 documents)==> message from rtsend, class  .........
```

Figure 2-61. sat_feed TOPIC Database Builder (dp1) Audit Log

2.10.4 TOPIC Partition Merger (mg1) Audit Log

The **sat_feed** has a TOPIC Partition Merger process, called **mg1,** that consolidates the real-time partitions built by **dp1** and writes it to its own audit log called **mg1.adt**. The **mg1** audit log is created, if not previously existing, when the process is first brought up and will continue to write to the same file until it is deleted and/or purged. Figure 2-62 has a sample of the audit information collected in the **mg1.adt** audit log file. This file can get very large so the System Administrator should watch the file size and purge it from time to time. The System Administrator will only look at this log file when “ghost” partitions appear; partitions that did not get deleted properly by TOPIC after they were merged.

```
rt - Version 3.1.5 (_ssol22, Rev C Mar 8 1994)
Logging in to server as mg1Logged in to server as mg1Level 0 (>= 1024 documents) (aaa)Level 1
(>=  256 documents) (aab)  _mg11aaa     264Level 2 (>=   64 documents) (aag)  _mg12aad
69  _mg12aae      64  _mg12aaf      64Level 3 (>=   16 documents) (aao)  _mg13aam      16  _mg13aan
16Level 4 (>=    4 documents) (acn)  _mg14ack       4  _mg14acl       4  _mg14acm       4Level 5
(>=    1 documents) (aaa)==> message from dp1, class NEWPARTReceived new partition _mg1xatl
(1 documents)==> message from dp1, class NEWPARTReceived new partition _mg1xatm (1 documents)==>
message from dp1, class NEWPARTReceived new partition _mg1xatn (1 documents)==> message from dp1,
class NEWPARTReceived new partition _mg1xato (1 documents)
Merging: _mg14ack (4 documents) _mg14acl (4 documents) _mg14acm (4 documents) _mg1xatl
(1 documents) _mg1xatm (1 documents) _mg1xatn (1 documents) _mg1xato (1 documents)Into: (_mg13aao)
Creating partition _mg13aaoMerging the document indexes
Merging with _mg14ack.did (98 entries, 4 documents)Merging with _mg14acl.did (101 entries, 4 documents)
Merging with _mg14acm.did (129 entries, 4 documents)Merging with _mg1xatl.did (70 entries, 1 documents)
Merging with _mg1xatm.did (176 entries, 1 documents)Merging with _mg1xatn.did (173 entries, 1 documents)
Merging with _mg1xato.did (97 entries, 1 documents)Entry 100) 128Entry 200) DECEntry 300) RUSNJPL0140
Merging case-insensitive indexes  writing record 250Merging stem indexes  writing record 250
Merging soundex indexes  writing record 250Writing accelerated indexMerging the topic indexes
Merging with /h/AMHS/Server/topic/amhs_db/systopic/sysind/_mg14ack.sid
Merging with /h/AMHS/Server/topic/amhs_db/systopic/sysind/_mg14acl.sid
Merging with /h/AMHS/Server/topic/amhs_db/systopic/sysind/_mg14acm.sid
Merging with /h/AMHS/Server/topic/amhs_db/systopic/sysind/_mg1xatl.sid
Merging with /h/AMHS/Server/topic/amhs_db/systopic/sysind/_mg1xatm.sid
Merging with /h/AMHS/Server/topic/amhs_db/systopic/sysind/_mg1xatn.sid
Merging with /h/AMHS/Server/topic/amhs_db/systopic/sysind/_mg1xato.sid
Merging 24 topicsMerging the document datasetsMerging with _mg14ack/_mg14ack.ddd
Merging with _mg14acl/_mg14acl.dddMerging with _mg14acm/_mg14acm.ddd
Merging with _mg1xatl/_mg1xatl.dddMerging with _mg1xatm/_mg1xatm.ddd
Merging with _mg1xatn/_mg1xatn.dddMerging with _mg1xato/_mg1xato.ddd
Partition _mg13aao prepared==> message from server, class DELETE-PARTITION
Deleting partition _mg14ackDeleting partition _mg14aclDeleting partition _mg14acm
Deleting partition _mg1xatlDeleting partition _mg1xatmDeleting partition _mg1xatn
Deleting partition _mg1xatoFinished delete ........
```

Figure 2-62. sat_feed TOPIC Partition Merger (mg1) Audit Log

2.10.5 CBC_FEED Audit Logs

When the **cbc_feed** is running, **cbc_feed** logs are generated daily and named with the Julian date, e.g. cbc_001.log, cbc_002.log ... cbc_365.log. The **cbc_feed**, like the **sat_feed** and its TOPIC processes (**dp1**, **mg1**), has audit logs as well. The TOPIC Database Builder for the **cbc_feed** is **dp4** and the Partition Merger is **mg4**. The audit files are so similar to the above examples they will not be included here. However, the **cbc_feed** does collect information that may be useful to the System Administrator. Figure 2-63 shows a startup error condition and the kind of error message that can occur if the initialization files, **vardef** and **daclist**, are not in synchronization. Figure 2-63 also includes normal auditing of a message that was successfully fed into the TOPIC database.

The System Administrator will reference these audit logs when a customer makes a request or needs to trace a message out of the system but should check daily for any ERROR entries. If an error condition occurred the audit log will indicate that the message was moved to the **cbc_feed** error directory in the AMHS Real-Time directory and why (the directory is located: **/h/AMHS/Server/topic/amhs_db/error/cbc_feed**).

NOTE: The System Administrator should check the **/h/AMHS/Server/topic/amhs_db/error/cbc_feed** directory daily to make corrections where needed and feed the message into the database.

```
cbc_feed started Dec 6 11:25:41 Fatal Error:
cbc_feed: Unable to initialize DAC information.
Vardef error occured attempting to access <$special_dir
>Invalid: directory not specified <$special_dir>.
Program aborted

cbc_feed started Dec6 11:30:00
Dec 6 11:30:01 INFO:   cbc_feed:
  Received: /h/AMHS/Server/sat/autodin/archive/t333/d011204.msg
  Precedence:  Z  Class:  U
  Format line 2:
    ZCAUZYUW RUSNJPL1357 3330112 MTMT-UUUU--RUSNASA.
  Acess type:  General
  TOPIC Mask: 0x00ffff00
Dec 6 11:30:02 INFO:   cbc_feed:  Sent to Topic: /h/AMHS/Server/dac/general/r334/d011204.msg
```

Figure 2-63. cbc_feed Audit Log

2.10.6 TOPIC Central Profiler (pf1) Audit Log

The TOPIC Central Profiler can be made up of several profiling processes. Each one is uniquely named, and typically the site is delivered with a **pf0** and **pf1** profiler. The Central Profiler audit logs will be of interest to the System Administrator for purposes of determining who in their customer base is receiving messages in their queues.

In Figure 2-64, the Topic Profiler is called **pf1**. The audit logs for all profilers will consist of the same information but will be identified within the file with their process ID (line 2 of Figure 2-64). Observe the notification of new partitions received by the **pf1** process and the number of messages in each partition. The profiles file for the **pf1** process contains a list of queries associated with TOPIC user accounts that map to the ACTION, INFO, and MSGS_SENT TOPIC message queues. Each query in the file is executed against the new partition and when a "hit" occurs, a system call is made with a program called "profile_hit".

```
rt - Version 3.1.5 (_ssol22, Rev C Mar 8 1994)
Finished compiling profiles Logging in to server as pf1==> message from server, class PARTTAB-RESPONSEID
Received partition _mg11aaa from server (264 docs)
Received partition _mg42aaa from server (64 docs)Received partition _mg12aad from server (69 docs)
Received partition _mg12aae from server (64 docs)Received partition _mg12aaf from server (64 docs)
Received partition _mg43aad from server (16 docs)Received partition _mg13aam from server (16 docs)
Received partition _mg43aae from server (16 docs)Received partition _mg43aaf from server (16 docs)
Received partition _mg13aan from server (16 docs)Received partition _mg44aav from server (4 docs)
Received partition _mg14ack from server (4 docs)Received partition _mg44aaw from server (4 docs)
Received partition _mg14acl from server (4 docs)Received partition _mg44aax from server (4 docs)
Received partition _mg14acm from server (4 docs)Received partition _mg4xaeu from server (1 docs)
Received partition _mg4xaev from server (1 docs)Received partition _mg4xaew from server (1 docs)
Received 19 partitions from server==> message from server, class NEWPARTPartition _mg1xatl prepar
ed (1 documents)==> message from server, class NEWPARTPartition _mg1xatm prepared (1 documents)=
=> message from server, class NEWPARTPartition _mg1xatn prepared (1 documents)
Making system call: /h/AMHS/Client/progs/profile_hit 0 Autodin GCCS__GCCSUSER '/h/AMRetrieval
 Summary for Partition _mg1xatl  Profile: <FILTER>(DOCSOURCE = COMEBAK) AND GCCS__ISSO-c
 Profile: <FILTER>(DOCSOURCE = AUTODIN) AND GCCS__ISSO-i
 User: GCCSISSO  Hits: 1
 Profile: <FILTER>(DOCSOURCE = AUTODIN) AND GCCS__ISSO-a
 Profile: <FILTER>(DOCSOURCE = COMEBAK) AND GCCS__GCCSUSER-c
 Profile: <FILTER>(DOCSOURCE = AUTODIN) AND GCCS__GCCSUSER-i
 Profile: <FILTER>(DOCSOURCE = AUTODIN) AND GCCS__GCCSUSER-a
 Profile: <FILTER>(DOCSOURCE = COMEBAK) AND amhs_dba__amhs_dba-c
 Profile: <FILTER>(DOCSOURCE = AUTODIN) AND amhs_dba__amhs_dba-i
 Profile: <FILTER>(DOCSOURCE = AUTODIN) AND amhs_dba__amhs_dba-a==>
 message from server, class NEWPARTPartition _mg1xato prepared (1 documents)
 Making system call: /h/AMHS/Client/progs/profile_hit 0 Autodin GCCS__GCCSUSER '/h/AMRetrieval
 Summary for Partition _mg1xato  Profile: <FILTER>(DOCSOURCE = COMEBAK) AND GCCS__ISSO-c
 Profile: <FILTER>(DOCSOURCE = AUTODIN) AND GCCS__ISSO-i
 User: GCCSISSO  Hits: 1
 Profile: <FILTER>(DOCSOURCE = AUTODIN) AND GCCS__ISSO-a
```

Figure 2-64. TOPIC Central Profiler (pf1) Audit Log

The System Administrator will reference this log when messages are not getting profiled to the appropriate user. This file can get very large very quickly, so the System Administrator should watch the file size.

2.10.7 TOPIC Server Audit Log

The TOPIC Server process (server) records the login and logout of all processes run in the TOPIC Real-Time System. The System Administrator would only look at this file to determine why other TOPIC processes are failing. See Figure 2-65.

```
rt - Version 3.1.5 (_ssol22, Rev C Mar 8 1994)==>
 message from pf0, class LOGINReceived login from (pf0), ID=6040==
> message from pf1, class LOGINReceived login from (pf1), ID=6043==
> message from pf0, class PARTTAB-REQUEST==> message from pf1, class PARTTAB-REQUEST==
> message from mg1, class LOGINReceived login from ==> message from pf0, class PARTTAB-REQUEST==
> message from pf1, class PARTTAB-REQUEST==> message from pf0, class LOGOUTReceived logout from (pf0)=
=> message from dp1, class NEWPARTReceived new partition message
 (_mg1xatl)Partition state updatedBroadcasting to all usersDone processing message
==> message from dp1, class NEWPARTReceived new partition message
 (_mg1xatm)Partition state updatedBroadcasting to all usersDone processing message
==> message from dp1, class NEWPARTReceived new partition message
 (_mg1xatn)Partition state updatedBroadcasting to all usersDone processing message
==> message from dp1, class NEWPARTReceived new partition message
 (_mg1xato)Partition state updatedBroadcasting to all usersDone processing message
==> message from mg1, class MRG[DEL]PARTReceived replace partition message
 (_mg13aao) 0) _mg14ack 1) _mg14acl 2) _mg14acm 3) _mg1xatl 4) _mg1xatm 5) _mg1x
atn 6) _mg1xatoPartition state updatedBroadcasting to all usersQueueing delete of old pa
rts for 120 secsDone processing message ......
```

Figure 2-65. TOPIC Server Audit Log

2.11 USER ACTIVITY AUDITING

2.11.1 AMHS TOPIC Client Audit Logs

User access to the TOPIC message database is always audited. There are individual audit files created for each TOPIC user account. The naming convention for these files includes the TOPIC user account name, found in the TOPIC **password** file, followed by a .tlg which stands for “TOPIC log”. For instance, the AMHS Database Administrator TOPIC account is **amhs_dba** and the audit log file for this account is **amhs_dba.tlg**. The **amhs_dba** account will be used to build topics for users as well as for deleting messages from the database. All actions of the

amhs_dba will be audited. These audit logs are kept in **/h/AMHS/Server/dac/topic_audits** directory (see Figure 2-66). TOPIC users are audited for log on and log out of the message database, retrievals against the database, clearing messages from their queues, printing of messages, and viewing messages. Additional flags can be set but audit files get very large very fast (auditing is turned on in the **/h/AMHS/Server/topic/amhs_db/sysfile/master.prf** file).

```
Jan 24 17:36:11 1996 0 <<Start>> Session
Jan 24 17:36:33 1996 1 <<Start>> Start Retrieval Topic: '<<None>>', Source: '<<None>>'
Jan 24 17:36:33 1996 2 <<Start>> Start Retrieval Topic: '<<None>>', Source: '<<None>>'
Jan 24 17:36:33 1996 3 <<Start>> Start Retrieval Topic: '<<None>>', Source: '<<None>>'
Jan 24 17:36:34 1996 4 <<Start>> Start Retrieval Topic: '<<None>>', Source: '<<None>>'
Jan 24 17:36:34 1996 5 <<Start>> Start Retrieval Topic: '<<None>>', Source: '<<None>>'
Jan 24 17:36:34 1996 6 <<Start>> Start Retrieval Topic: '<<None>>', Source: '<<None>>'
Jan 24 17:36:56 1996 7 <<Start>> Open Document 1001109 /h/AMHS/Server/dac/pers/r024/2400200
Jan 24 17:37:12 1996 7 <<End>> Open Selection 1001109 /h/AMHS/Server/dac/pers/r024/2400200
Jan 24 17:38:01 1996 8 <<Start>> Open Document 1001109 /h/AMHS/Server/dac/pers/r024/2400200
Jan 24 17:38:06 1996 8 <<End>> Open Selection 1001109 /h/AMHS/Server/dac/pers/r024/2400200
Jan 24 17:44:53 1996 9 Get Info Document Document
Jan 24 17:46:12 1996 5 <<End>> Start Retrieval Retrieved 1495 out of 1495 processed (100/100)
Jan 24 18:36:05 1996 0 <<End>> Session
```

Figure 2-66. AMHS TOPIC Audit Logs

The System Administrator will only look at these logs in the event a customer has some questions about their queues, but more likely the System Security Officer will be interested in these in the event of a security breach or compromise. The audit files for individual users will become very large, so the System Administrator needs to watch the file sizes.

2.11.2 AUTODIN Delivery Records

Another user audit log is the delivery record that is created for both inbound and outbound messages. For inbound messages the delivery record contains the discretionary access group, classification, precedence, originator, file path, time of receipt, DTG, and the list of message recipients by Topic account. Figure 2-67 is a typical AUTODIN inbound message delivery record.

```
!--Delivery Record--------------------------------------------------------------------------------
!Discretionary Access:          General
!Classification:                R
!Precedence:                    R
!Originator:                    JPL
!File Path:                     /h/AMHS/Server/dac/general
!Time of Receipt                061719 Z Dec 95  366/310011
!DTG:                           310011 Z Dec 95
!Delivered to:
        amhs_oper
```

Figure 2-67. AUTODIN Inbound Message Delivery Record

2.11.3 Comeback Message User Record

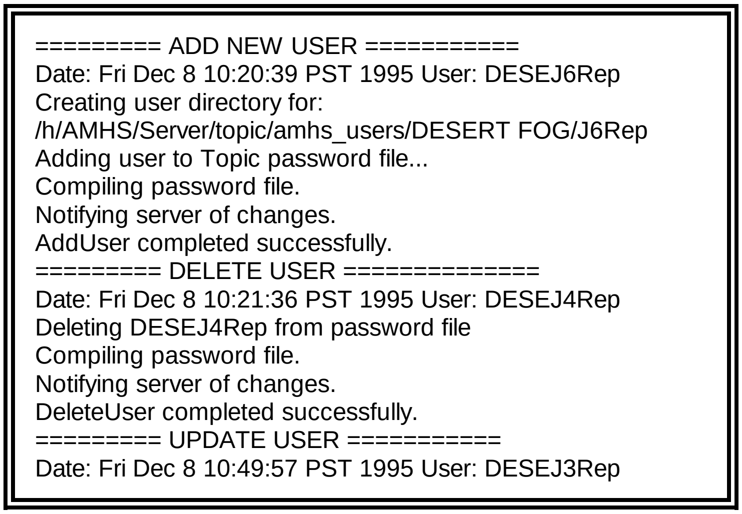
The outbound message (comeback copy) delivery record contains the information collected during message coordination. It includes the message Drafter (userid) and remarks, each Coordination Point (userid) and remarks, and the Message Releaser (userid) and remarks as well as the time stamp of when the message was released. Figure 2-68 shows a comeback copy with message delivery record.

```
RCASZYUW RUSNJPL0519 0042327 MTMS-SSSS--RUSNSUU.
ZNY SSSSS
R 042326Z JAN 96
.FM JPL PASADENA CA
.TO CINCUSAREUR HEIDELBERG GE //GCCSUSER//
.INFO USCINCEUR VAIHINGEN GE //AMHSPOS1//
.BT
S E C R E T
SUBJ: FROM MM - 173 - FALSE TRUE . . . . .
1
2
3
4
5
6
7
8
9
10
DECL/070197//
BT
RCASZYUW RUSNJPL0519 0042327 MTMS-SSSS                              NNNN

! DAC: General
************************************************************
!DRAFTER:  willie   GCCSUSER           DTG: 230012Z DEC 95
------------------------------------------------------------



************************************************************
!RESPONSE BY: willie   GCCSUSER         DTG: 230029Z DEC 95
------------------------------------------------------------

************************************************************
!RELEASED BY: willie   GCCSUSER DTG: 042325Z JAN 96
```

Figure 2-68. Outbound Message (Comeback Copy) Delivery Record (MAR)

2.11.4 Creating TOPIC Accounts Audit Log

This audit log contains information about when new AMHS TOPIC accounts are created, changed or deleted. It has the date time stamp and user ID. The System Administrator will use this log to identify when a user account was created. See Figure 2-69.



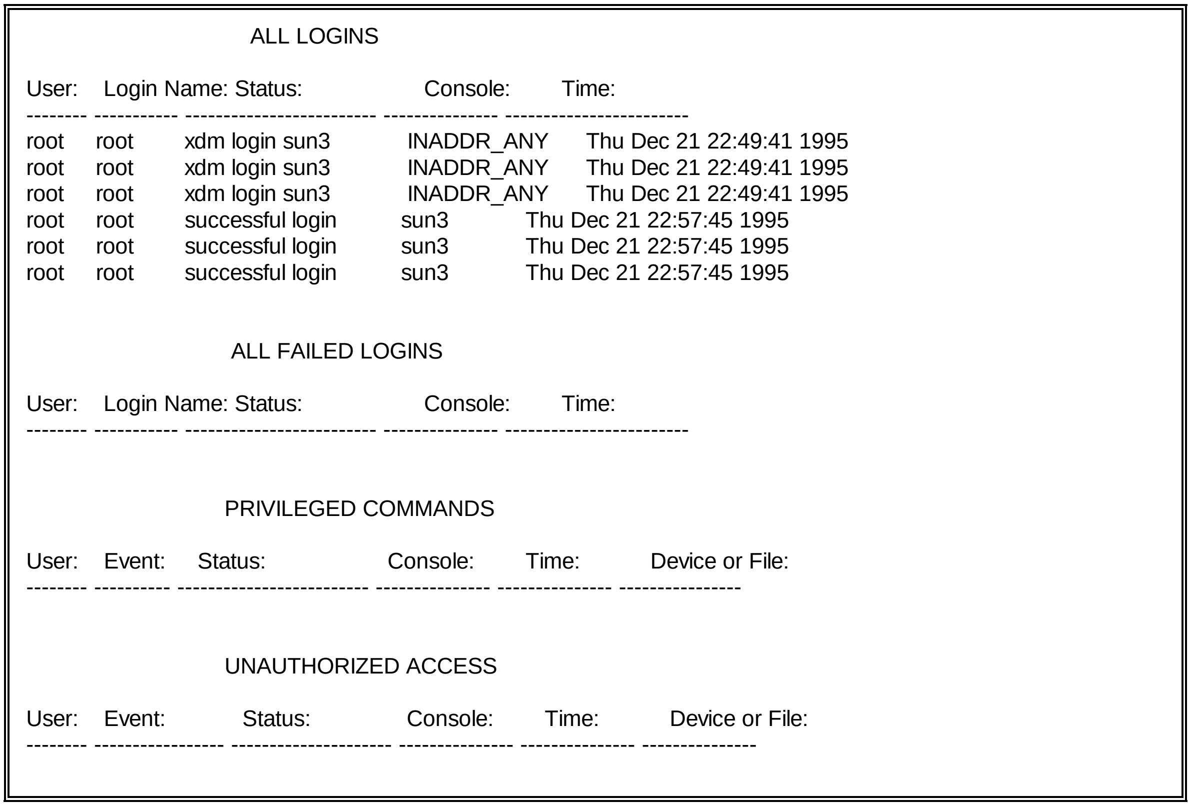
```
========= ADD NEW USER ===========
Date: Fri Dec 8 10:20:39 PST 1995 User: DESEJ6Rep
Creating user directory for:
/h/AMHS/Server/topic/amhs_users/DESERT FOG/J6Rep
Adding user to Topic password file...
Compiling password file.
Notifying server of changes.
AddUser completed successfully.
========= DELETE USER ==============
Date: Fri Dec 8 10:21:36 PST 1995 User: DESEJ4Rep
Deleting DESEJ4Rep from password file
Compiling password file.
Notifying server of changes.
DeleteUser completed successfully.
========= UPDATE USER ===========
Date: Fri Dec 8 10:49:57 PST 1995 User: DESEJ3Rep
```

Figure 2-69. New User Audit Logs

2.11.5 Solaris BSM Audit Report

The Security Manager software from the Executive Manager (EM) segment provides the capability for viewing certain events on the system, and the ability to print out the results by date range. The audit files available to the Security Manager Audit Report feature are only those which have been downloaded from the client workstations to the EM Server; audit files are downloaded when a 24-hour period has passed or the workstation has been rebooted. The types of events to view are all logins and logouts, including failed logins, unauthorized access attempts, and privileged commands (e.g. change read, write, execute, etc.). These classes of audit events are defined in a Solaris configuration file in **/etc/security** called **audit_control**. If other accounts, used to administer the GCCS AMHS like "**secman**" or "**sysadmin**" or "**root**", need to be viewed, the Solaris BSM utilities such as **auditreduce** and **praudit** can be used; Solaris provides two routines to view audit files: **auditreduce** filters selected events from the audit files and **praudit** takes the output from **auditreduce** and puts it into more readable form. See Figure 2-70.

Most likely the System Security Officer will be interested in the report in the event of a security compromise. The System Administrator/System Security Officer needs to remember that the data available for viewing is 24 hours old if the workstations have not been rebooted.



```
                         ALL LOGINS

User:   Login Name: Status:              Console:       Time:
-------- ----------- ------------------------ --------------- ------------------------
root    root      xdm login sun3          INADDR_ANY     Thu Dec 21 22:49:41 1995
root    root      xdm login sun3          INADDR_ANY     Thu Dec 21 22:49:41 1995
root    root      xdm login sun3          INADDR_ANY     Thu Dec 21 22:49:41 1995
root    root      successful login        sun3           Thu Dec 21 22:57:45 1995
root    root      successful login        sun3           Thu Dec 21 22:57:45 1995
root    root      successful login        sun3           Thu Dec 21 22:57:45 1995


                       ALL FAILED LOGINS

User:   Login Name: Status:              Console:       Time:
-------- ----------- ------------------------ --------------- ------------------------


                       PRIVILEGED COMMANDS

User:   Event:    Status:              Console:       Time:          Device or File:
-------- ---------- ------------------------ --------------- --------------- ----------------


                       UNAUTHORIZED ACCESS

User:   Event:         Status:            Console:       Time:          Device or File:
-------- ---------------- --------------------- --------------- --------------- ---------------
```

Figure 2-70. BSM Audit Reports

2.12 LOG MAINTENANCE

The AMHS maintains a number of logs. These logs can accumulate a lot of information over time, reducing the available disk space for on-line messages. Site-specific SOPs need to be developed and approved to establish policies for log maintenance.

3. AMHS BASELINE CONFIGURATION

In support of the GCCS mission, the AMHS facilitates the free flow of information with the constraints of operational security. The need for information to be user specific in some cases, available for multishifts in others, or shared between teams creates unique challenges for the AMHS and the people who support it. In Section 2, you developed an understanding of how the AMHS functions and an appreciation of its capabilities. In this section you will develop the skills necessary to plan, configure and deploy the system, exploiting the capabilities of the AMHS to the maximum benefit of the users.

The AMHS has been DMS certified as a system based on certain assumptions. The system will be installed, configured, operated and maintained consistent with the Site Description document and pre-approved policies. Whenever new installations, reinstall upgrades or configuration modifications are planned, be sure the plans have the approval of both the cognizant Security Officer and the local GCCS Approval Authority.

This section is a road map for planning, implementing, operating and maintaining the GCCS AMHS system. You will cover the concept and be pointed to processes and procedures in other sections of this document and other documents for the specific steps. This is consistent with the concept of an evolving system. Many of the underlying philosophies of the GCCS program are drawn from the principles of Total Quality Management (TQM), Paradigm shifts and continuous improvement. The modular approach allows pieces to be upgraded, improved and/or modified as the need arises or technology shifts without scrapping significant portions of the system.

The primary documents you will be using are:

CM # LL-500-L03-19	GCCS Integration Support Implementation Procedure, Rev 1.
CM # LL-500-29-12	Joint Interoperability And Engineering Organization System Administration Manual, GCCS-SAM-2.1
CM # LL-500-XXX-XX	GCCS Install Segment And Update Tape Release Bulletins
	JPL D-12797, AMHS System Administration Manual
CM # N-TOP-01-01	Topic Database Administration Guide, V 3.1

3.1 SYSTEM ISSUES

The first step is planning. If this is a new AMHS install, the hardware must be in place, connected to the network, operating system software installed, and the UNIX environment tested for functionality. The EM Server must be installed and operational. Pages B-2 thru B-7 of Appendix B describe specific steps. If this is an upgrade, B-2 thru B-7 still apply.

The AMHS support team needs to know the system and the users. The GCCS Operator and AMHS Systems Administrator are typically the first point of contact for AMHS users with system problems. They have the responsibility to ensure that the AIS is always available and meeting user needs. One suggestion might be to color code the organization chart, marking all GCCS users and highlighting the AMHS users. Another is to set up databases for Configuration Management, user information, TOPIC Query trees and test messages. These could be set up using Applix spreadsheets or MS Excel. There are several important sections in the CM # LL-500-103-19 Implementation Procedures; Sections 2, 3 , 4 and Appendices H and I. The GCCS Ver 2.1 Segment Tape and all the update tapes have release notes that must be reviewed for AMHS-specific information to include in your planning. The CM# LL-500-29-12 GCCS-SAM-S.1, Sections 10.1, 17.3, and 19.1 through 19.4, address AMHS issues. Section 3 describes the Segment Install Process. All of these areas need to be reviewed and discussed with the GCCS System Administrator as part of the planning process.

Next, review the security procedures and SOPs for your site and perhaps get copies to AMHS-specific SOPs for other existing sites. Most site SOPs evolve into proven techniques over time. Develop a method for collecting user requirements for AMHS TOPIC profiling and other user specific configuration information. Section 2.4 covers several concepts such as dead letters and a user topic query demonstration. With the completion of the planning and information gathering phase, the AMHS hardware operating system should be tested, the EM Server operational, and the AMHS Server and dedicated workstation should perform as an EM Client with the standard desktop and suite of services.

At this time supply the GCCS ISSO or proper approval authority with a list of users that need AMHS accounts. Most will already have GCCS accounts, but you need to supply this information because an AMHS user may need additional information added to their EM configuration.

GCCS Account Fields:

8	Host Name	AMHS SRVR
8	userid	TCJ63DOE
25	User Name	Maj_John_Doe_TCJ63,X4-666
25	Project Name	Current_Operations
8	Position Name	Sys Admin
25	Position Description	GCCS Ops Manager
25	Directorate	Information System J6
25	Division	
25	Branch	
25	Section	
25	Cell	

The last five categories are available to create working groups, people with a need to communicate and work as a team, that can easily share folders and buckslips.

Once approved, add this information to the user database for reference, hard copy in a notebook or add to the spreadsheet.

The AMHS Server and any new AMHS support workstations should have all the standard GCCS software installed and functioning normally.

Once the planning phase is complete , the AMHS pre-install phase starts.

3.2 PRE-INSTALL PHASE

The pre-install phase involves adding to the EM Server several usernames and an initial set of security settings, including adding the AMHS Server to the NIS+ services and installing the Sun Solaris BSM software component. The SAT is also set up to be network aware with PCNFS. Appendix B, Section V, Pages B-4 through B-11, contains detailed steps.

A loop back plug is needed later for the SAT to test message flow and SAT - AMHS interfaces.

The pin outs are:

DB25F with pins 2 to 3 jumpered and 15, 17 ,24 jumpered.
SAT pin outs are described in Appendix D of the SAT Manual.

With this plug in place the outgoing messages are looped back to the SAT input and received as incoming traffic. This facilitates local testing of the AUTODIN interface.

3.3 AMHS SERVER SOFTWARE INSTALLATION

The AMHS Server software install process is included in Appendix B, Pages B-7 through B-11.

Segments required for a new load:

Order	Name	Date	Version
1	GCCS COE		X
2	EM V2.1 Upgrade	08/15/95	2.1.6
3	EM Printer Admin	01/12/96	2.1.9
4	Applix 3.2	01/12/95	3.2
5	Cmd Ctr Apps	08/05/95	2.1.5
6	CCAPPS AMHS Patch	10/25/95	1.0
7	COTS Topic	12/02/95	3.1.5c
8*	AMHS Server	08/05/95	2.1.4
9	AMHS Client	08/05/95	2.1.4
10	CCAPPS MM Patch	12/21/95	1.3
11	EM Launch Patch	12/27/95	1.1
12*	AMHS Server Patch	12/10/95	3.1
13	AMHS Client Patch	12/09/95	3.1

Figure 3-1. Segments Required For A New Load

Segment required for an upgrade load (it is assumed the AMHS is already installed):

Order	Name	Date	Version
1	GCCS COE		X + 1
2	EM Printer Admin	01/12/96	2.1.9
3	CCAPPS Overload	05/01/95	1.0
4	Cmd Ctr Apps	08/05/95	2.1.5
5	CCAPPS AMHS Patch	10/25/95	1.0
6	AMHS Server Upgrade	08/05/95	1.0.1
7	AMHS Client	08/05/95	2.1.4
8	CCAPPS MM Patch	12/21/95	1.3
9	EM Launch Patch	12/27/95	1.1
10*	AMHS Server Patch	12/10/95	3.1
11	AMHS Client Patch	12/09/95	3.1

Figure 3-2. Segments Required for an Upgrade

* Install on server workstation only.

The list for an AMHS Client is the same as the list for an AMHS Server with two exceptions. DO NOT load the AMHS Server Segment and the AMHS Server Patch on a client. Keep in mind that this listing of segments is required for an AMHS and not for any other GCCS functionality (i.e., JMCIS, Netscape, GRIS, Auditing, etc.). There are two lists, one for installing new workstations from scratch and the other for upgrading existing workstations. The AMHS Server console includes client software for operation and testing. As mentioned earlier it is important to review all the segment upgrades. For example, the AMHS Sys Admin Tools, described in Section 5, are on one of the update segment tapes.

3.3.1 SAT Installation and Configuration

The SAT is installed and operated consistent with the procedures described in Cavalier Communication Inc. Standard Automated Terminal System User's Manual. This document covers both the installation and operation of the SAT. The installation of the SAT as part of the AMHS system is covered on pages B-13 through B-15 followed by a startup procedure and refers to the test procedure presented in the JPL AMHS installation class material.

3.3.2 SAT Configuration Files

Appendix B, Pages B-14 through B-21, list SAT installation specific configuration files. The SAT is DMS Certified as a stand-alone AUTODIN terminal, and the AMHS interface maintains that certification. The system has been accepted as cooperating components.

3.4 CLIENT WORKSTATION INSTALL

The segment tapes have an option for client or server installs. The client install setups are the same as for server workstations except those items noted with an asterisk (*) in Figures 3-1 and 3-2. Each workstation must be NFS mounted to the server after installation. This needs to be checked to be sure the segment install handled this correctly. With the NFS Mounts in place, the workstation and server file structure will appear to be very similar. Figure 2-56 shows this in detail. Pages B-12 and B-13 are the client installation steps.

3.4.1 Post Installation

The user account must be set up, including the AMHS launch icons for Message Manager, MTF Editor, and AMHS topic query tools.

3.4.2 Software Updates

The AMHS software will be updated from time to time via patch segments supplied by DISA, which will include installation and operation notes detailing the changes and how they affect system operation.

3.5 SYSTEM ACCEPTANCE TESTING

System acceptance testing has two phases to it.

(1) DMS Certification compliance. The AMHS is certified as a system (**JPL D-xxxx, AMHS TEST PLAN and JPL D- xxxx, Site description)** but requires that the site demonstrate the specific installation conforms to certification standards. Appendix B, Pages B-31 through B-38, describes the test procedure to be performed at each site.

(2) System tests configured to meet the specific needs of the users. This is an ongoing testing requirement because user needs change and Mission requirements change. A well designed set of test messages and the loop back plug described in Section 3.2 are two required tools.

3.6 REDUNDANT AMHS SYSTEM INSTALLATION

Section 6 addresses the redundant system Theory of Operation and Administration, and Appendix B, Pages B-23 and 24, describes the installation and setup. Page B-25 includes procedures for testing the installed configuration.

3.7 USER ISSUES

Each user may feel their requirements are unique, but after developing skills in knowledge base engineering, part of the profile building process, it will become obvious that all AUTODIN traffic, and subsequently the users, have requirements in common.

Each incoming message must be delivered to the appropriate person or returned to the sender. Also, users other than the addressee may have a **need to know**, or the message may fall within their **areas of interest**, necessitating they see originals or copies of this traffic. The effective design and testing of user profiles will easily meet both of these requirements.

3.8 INFRASTRUCTURE ISSUES.

To effectively manage the AMHS system, detailed records must be maintained. Some concepts are discussed in Section 2.4.4 and the following form is an example of how one of these requirements might be accomplished.

AMHS User Message Delivery Questionnaire

- AUTODIN messages are distributed within the AMHS based on keywords or combinations thereof. As an AMHS Client you have an ‘ACTION’ and ‘INFO’ queue where messages can be delivered as well as access to the general database. If you would like messages delivered to you based on ACTION and / or INFO, list those keywords or phrases in the appropriate box below. If you would like delivery based on keywordsand your PLA/Office Symbol or other identifier, check the box next to that criteria.

Name: ______________________________

	Yes
PLA: ______________________________	□
Office Symbol: ______________________________	□
Other: ______________________________	□

ACTION keywords and/or phrases:

INFO keywords and/or phrases:

Please return this questionnaire to your AMHS Adminstrator for AMHS message delivery service.

If this is your first request be sure to plan to meet with the AMHS System Administrator.

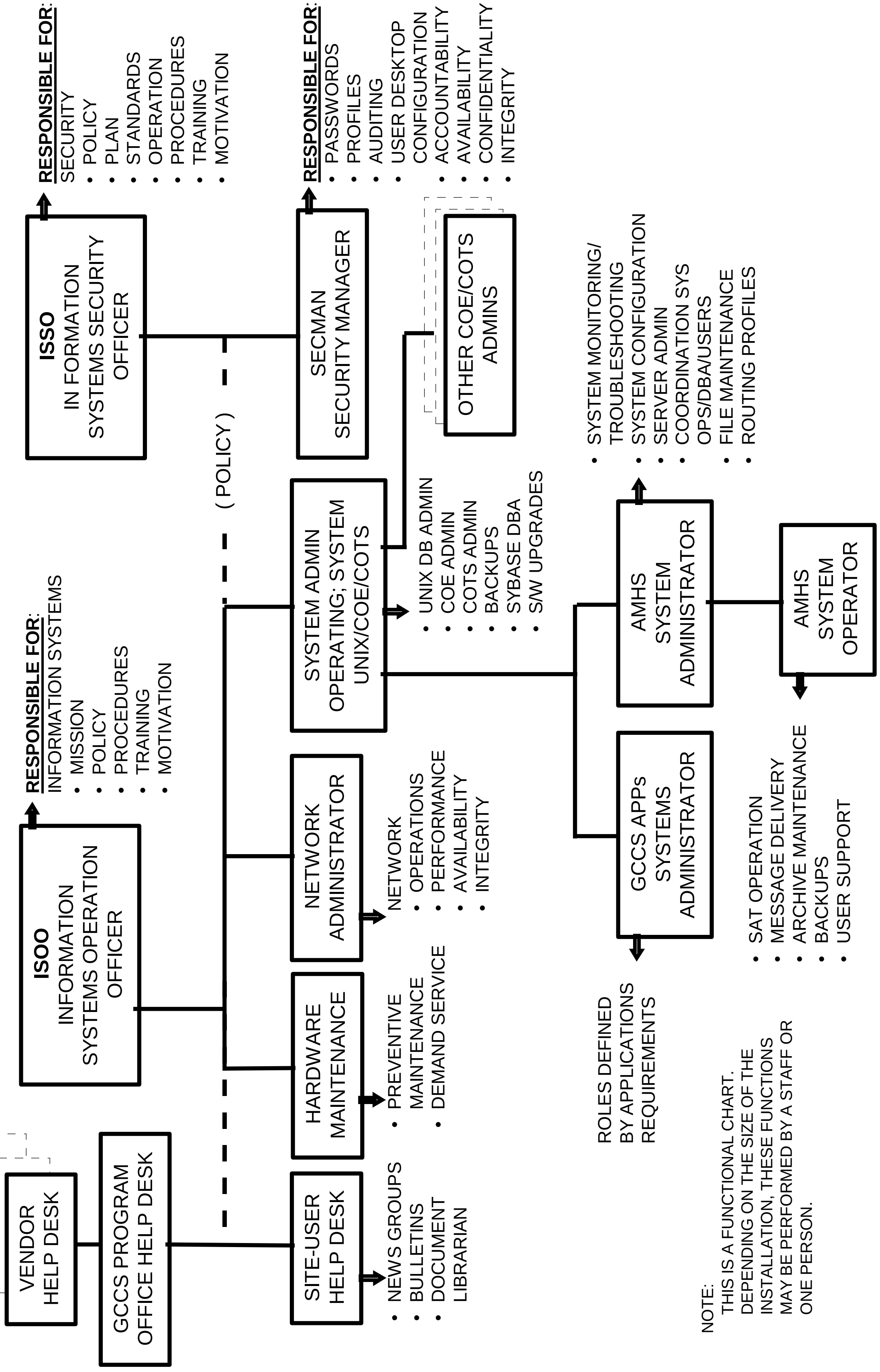
3.9 OPERATIONAL ISSUES

The AMHS is an extension of the AUTODIN DMS and requires special care in operation and maintenance to ensure compliance with mission requirements and DMS regulations. Every support feature built into the AMHS has to take these considerations into account. The UNIX environment, with its enhanced security and transaction logging enabled, gives clear audit logs. The AMHS and the Topic message processing engine also has audit logs, giving the System Administrator and Security personnel well thought out tools to assist with compliance. The other component of the system is the people, procedures and organizational support structure. The GCCS System Security Implementation Instructions for Site Security Administrators (LL-500_43-04) describes the Security Requirements and detailed procedures for compliance. The GCCS System and Network Management Conops v1.6 document (LL-500-189-01) describes in detail the GCCS System, how the AMHS fits in the plan, and most importantly the organizational support structure. Section 3.3.2, Page 29, describes the system environment:

> “The next suite of equipment is the Automated Message Handling System (AMHS). The system being used is the one produced by the Jet Propulsion Laboratory (JPL) for the Army. The system is uniquely identified as the JPL AMHS, GCCS Configuration. In actuality, three configurations will exist for the AMHS suite of equipment based on the hardware utilized for AMHS functionality. The first configuration consists of two SUN Sparc 20s. One serves as the AMHS server while the other is the AMHS dedicated client. The dedicated Sparc 20 client workstation represents the one used by the operator responsible for the AMHS. It should be noted that any of the GCCS workstations can be loaded with the AMHS client software. The second configuration consist of a single Sparc 20 and utilizes separate file systems on the data base server to complete the AMHS functionality. The final configuration uses only the Sparc 1000 or 2000 data server with no dedicated Sparc 20s. Instead, the entire AMHS functionality resides on the database server. The next device in the AMHS suite is the Secure Autodin Terminal (SAT) which contains a specialized communications card for interfacing with the AUTODIN message system. The SAT computer is an Intel 486-based computer operating with the Microsoft Disk Operating System (MS DOS). The next component in the AMHS suite is the cryptographic devices feeding data to the SAT. A large variety of cryptographic devices are used with each being site specific. The next device in the AMHS suite is one of seven different components (AMME, AFAMPE, LDMX, MDT, ASC, CSP, and AGMS) which are an integral part of the AUTODIN feed to ensure 100% message delivery. Finally, there is a 4800 bits per second (bps) AUTODIN circuit feed. There are a wide variety of AMHS configurations. Each configuration has to undergo certification testing by DMS certified testers. One stipulation of AMHS certification is that the GCCS LAN, all remote LANs, all backside LANS, all remote dial-in subscribers, and all dedicated circuit subscribers must be protected at the Secret-NOFORN classification level, otherwise certification will be denied at that GCCS site.”

The following organization chart is not CONOPS compliant but rather a template for the site-specific one required locally.

GCCS SUPPORT INFRASTRUCTURE



NOTE:
THIS IS A FUNCTIONAL CHART.
DEPENDING ON THE SIZE OF THE
INSTALLATION, THESE FUNCTIONS
MAY BE PERFORMED BY A STAFF OR
ONE PERSON.

This page intentionally left blank.

4. SYSTEM OPERATION TASKS / FUNCTIONS

The following functions and tasks performed by GCCS AMHS support personnel may require a number of operators and administrators, or may be performed by a single administrator depending on the site size and hardware configuration. As described in Section 2, the AMHS is a COE component of the GCCS and operates in concert with the GCCS COE EM Server and GCCS Desktop, and depends upon secman, password, UNIX groups, EM Security Manager and EM Profile Manager Project/Position Pairs. Low level auditing is handled within the Solaris operating system and high level auditing is done by the EM and AMHS core software and the TOPIC COTS to track system, user and message traffic activities. The AMHS operator activities will normally be performed by the GCCS Data Center Operator team. When system or user problems are identified that are beyond the scope of the operator's duties, they will hand the problem off to hardware maintenance, the LAN Administrator or the AMHS System Administrator, as appropriate. The GCCS operator task descriptions and associated functions are meant to be generic, and may or may not apply to all GCCS AMHS systems.

The GCCS System Operator is responsible for operating the listed servers and supports user workstation issues. The System Administration tools run on client workstations and assist the operator in monitoring and maintaining the AMHS. The AMHS Server and EM Server are loaded with Client software and the consoles are normally running as clients.

(1) The AMHS Server, typically a Sun SPARC 20.

(2) The GCCS EM/NIS+ Server, typically a Sun SPARCstation 1000.

(3) The AMHS SAT, a PC 386/486.

(4) The secondary AMHS Server, typically a Sun SPARC 20 (if this is a redundant system installation).

(5) SAT Printer for printing SAT audit logs.

The System Operator uses the EM Desktop Monitor and Control functions to monitor workstations, servers and process status. The AMHS administration tool is used to assist with some of these tasks. It is up to the GCCS Operations Manager, AMHS Administrator and GCCS operator to develop procedures to standardize site-specific methods.

The System Operator's duties may include:

(1) Monitoring system operations. The System Operator monitors the operational status of the Real-Time Processor, Server and SAT; maintains the system logbook; monitors and responds to system console messages; monitors the network and interface status; and monitors AMHS message traffic and critical processes.

(2) Controlling system operations. The System Operator maintains the system configuration and is responsible for system initialization and startup, normal operations and shutdown. The System Operator maintains time synchronization between the processors, services the SAT printer, reloads the crypto key as required, and recovers the system after any failures.

(3) Performing system backups and file maintenance. The System Operator performs scheduled incremental and full file backups, loads tapes and restores files as required, and maintains the message archive at the appropriate level.

(4) Performing routine operator level maintenance on the processors, communications equipment and the SAT printer. The System Operator requests maintenance support when necessary.

(5) Processing dead letter traffic. The System Operator routes undelivered messages to appropriate recipient and enters note in the logbook.

(6) Troubleshooting. The System Operator responds to user requests for assistance with system problems.

NOTE: The System Operator is not responsible for handling AUTODIN service messages. The Automated Multi-Media Exchange (AMME) will route service messages for the AMHS to the traffic service position at the AMME, where AMME operators will respond to them. If service messages are received at the AMHS, the operator will notify the AMME operator or AMHS System Administrator to determine proper disposition.

The system operators will have accounts on the EM Server Client desktop and on the AMHS Server desktop. The AMHS System Operator accounts should be set up by the GCCS Approval Authority with a unique userid (8 characters maximum, e.g. John), user name (e.g. John Jones, Tsgt 4-6213, etc.—60 characters maximum) and a Project/Position Pair Current Operations (Project), AMHS operator (Position) (8 characters maximum) with permissions (25 characters maximum) for all the DAC groups and topic accounts. Individual accounts are necessary for any of the operators on any shift to perform these assigned duties while maintaining audit log traceability. With these privileges the operator must use care to only perform those operations defined in the SOPs, such as processing dead letters and backups, because they have the permissions that could damage important operational settings.

The System Operator performs routine tasks on the AMHS servers: initiates and performs AMHS file backups, AMHS system startups and shutdowns, and performs monitoring functions on the AMHS hardware components. One very important aspect of the operator's task is to maintain a System Logbook.

4.1 MAINTAINING THE SYSTEM LOGBOOK

The AMHS System Logbook serves as the focal point for documenting system changes, adding and deleting user accounts and the status of the system components. It is also used as a historical record of system operation. The AMHS System Logbook is kept near the Server console to be readily available at all times.

The System Operator maintains the System Logbook by manually recording the following information in it:

(1) At least once a day, the operational condition of the system (determined as a result of monitoring system status).

(2) Software changes and scripts that affect the system configuration (needed to help reconstruct the system after a crash or other hardware or software failure).

(3) Status of backups and other routine tasks.

(4) Status changes (shutdown, startup, reconfiguration of hardware/software, etc.).

(5) Any user identified problems and dispositions.

(6) Changes to user IDs profile, etc.

(7) Any anomalies.

The historical information in the logbook is critical for recovering and communicating system changes. The last backup of the system needs to be restored after an unscheduled shutdown/failure.

4.2 SYSTEM STARTUP AND SHUTDOWN

A typical AMHS hardware environment consists of:

(1) An AMHS Server, Sun SPARC 20.

(2) An Executive Manager NIS+ Server, Sun SPARC 1000 or SPARC 20.

(3) An AMHS SAT, a PC 386/486.

(4) GCCS workstations with AMHS Client software installed.

(5) A redundant AMHS Server and/or a Redundant SAT PC, if required.

An orderly sequence of operations is required for the AMHS to bootup (or powerdown) correctly. The following describes the procedures to perform orderly AMHS startup and shutdown to ensure no loss of data.

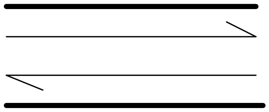
4.2.1 Starting the AMHS System

The AMHS machines should be started up in the following sequence.

(1) Ensure the GCCS EM Server is fully operational.

(2) Power-on all disk arrays and wait ten (10) seconds. (The ten seconds wait is very important.) Or follow the manufacturer’s recommendations if your system has third party drives.

(3) On systems with disk arrays, monitor the liquid crystal display (LCD) on the disk arrays for errors, and make sure all disks are up, as indicated by a darkened LCD bar for each installed drive. Ensure that the channel link indicator on the disk array LCD shows the fiber channel link as good. It should look like the following example:



(4) Switch on all monitors.

(5) Power-on the AMHS Server’s CPUs and note any errors.

(6) Log in to the **amhs_server** as the **amhs_dba** account.

(7) Type **topic_cmd** to enter the administrator commands menu.

(8) Select option **1** to **Start AMHS Processes**.

(9) Select option **0** for **Status**. Verify all the AMHS processes started with no problem.

(10) Power-on the SAT.

(11) Type the **amhs_dba’s** password when prompted. The SAT main menu will then automatically display.

(12) Select **Operator Login** .

(13) Type in the operator’s login and password.

(14) Select **Initialization Functions -> On Line** . The AMHS is now on-line.

4.2.2 Shutting Down the AMHS

The AMHS Server may be shut down either from the Session Manager Desktop or from a UNIX Xterm window. Either way, it is necessary to close all open data files and terminate all processes prior to system shutdown. **Method (2) is for emergencies only**.

(1) Shutting down AMHS from the UNIX Xterm window:

(a) Log off SAT. **The SAT should be put off-line to ensure that the AUTODIN switch holds the messages.**

(b) Login to the AMHS Server using the **amhs_dba** account.

(c) Type **topic_cmd** . (The result of this command is shown in Figure 4-2.)

(d) Enter option (**31**) to **Shutdown AMHS** .

(e) Enter option (**.**) Exit menu.

(f) From any Xterm <u>on each AMHS Server</u>, enter the following:

```
su  root
sync
sync                    (wait a few seconds)
cd  /
shutdown  -y  -g0  i0
```

(g) Wait for the **>OK** prompt on each server.

(h) Switch off both CPUs and SAT.

(i) Switch off all disk arrays.

(j) Switch off monitors.

NOTE: The SAT PC should be shut down with the servers, since it would have to be restarted anyway.

(2) Shutting down the AMHS via Session Manager: **(Method (1) is preferred)**

(a) Perform operator logoff on SAT. **The SAT should be put off-line to ensure that the AUTODIN switch holds the messages**. Wait for all the internal AMHS processes to finish processing the last message.

(b) From top menu select **System -> Shutdown**.

(c) It is now safe to powerdown the AMHS Server.

(d) Switch off all disk arrays (if applicable).

(e) Switch off SAT PC.

(f) Switch off monitors.

4.2.3 Shutting Down and Starting Up User Workstations

The System Operator will tell a user how to shut down and restart their workstation when requested to do so.

(1) To reboot the system (workstation) from the desktop with any login, use the pull-down menu at the top of the desktop and select: **System -> Restart**.

The system will now reboot itself.

(2) To shut down or bring up the system (workstation) from the desktop, do the following:

(a) To shut down, from the desktop select: **System -> Powerdown**. The system will shut down in a few moments, then:

1) Switch off power on the external drive(s).

2) Switch off power on the other peripherals.

3) Switch off power on the CPU.

4) Switch off power on the monitor.

(b) To powerup:

1) Switch on power to the peripherals.

2) Switch on power to the external drive(s).

3) Switch on power to the CPU.

4) Switch on power to the monitor.

The system should now boot up on its own. Log in as normal.

4.3 PROCESSING UNDELIVERED MESSAGES

The AMHS System Administrator will profile all messages that do not meet an existing profile to the AMHS operator position. These messages will be reviewed by the System Operator on duty (logged in on the AMHS Server) and routed to the appropriate recipient using a Message Manager buckslip. The event is logged in the AMHS logbook and, if the message is not unique, the buckslip comment field should instruct the recipient to request the AMHS Administrator to update his topic profile criteria to ensure similar future messages are properly routed.

Some sites may choose to identify someone other than the System Operators to receive and process these messages. If the profiling criteria identified by the users is well thought out, there should be very few of these messages.

Once per day the System Operator should run this script on each **dacdir** for yesterday's Julian day. This will look for messages that have not been delivered to anyone.

(1) **cd /amhs/dac/{dacdir}/d{jday}**
(Change directory to yesterday's Julian day for each **dacdir**.)

(2) This is a sample listing of the directory.

```
020003      211845      211845.001  211845.003
212130.004  212131      212204      211840
211845.000  211845.002  211908      212130.005
212137      212207
```

This represents a listing of all the delivery records on a certain Julian day.

(3) Execute the following commands from the csh.

```
foreach d (*)
? if (`tail -1 $d | cut -c1-1` == \!) then
?    echo "Not profiled: $d"
? endif
? end
```

The question marks "?" are provided by the shell.

(4) Sample output.

```
Not profiled: 020003
```

Message 020003 did not get profiled to a user.

There should not be any undelivered messages in any of the **dacdirs** if the system is functioning correctly. Your site System Administrator could expand this script to automatically scan each **dacdir** for yesterday's Julian date and route the results to his station with a crontab run at midnight.

These two processes ensure that the AUTODIN messages are all properly processed and dispositioned. An AMHS Server failure or SAT failure would cause the SAT/AMME handshake to be broken, holding the messages at the switch.

4.4 PREVENTIVE MAINTENANCE

A site-specific preventive maintenance plan and schedule should be developed to ensure maximum system availability. Cleaning fans and equipment, hard disk diagnostics, and similar activities are necessary and are defined as preventive maintenance tasks in the manufacturing documentation.

4.4.1 Procedures for Laser Printer Maintenance

Keep the Sun Postscript and HP LaserJet 4/4M printers' exterior and interior surfaces clean. Replace the toner cartridge and load paper as needed. (Refer to the SPARCPrinter II Hardware Installation and User Guide, Sun Part No: 801-5806-10, or the HP LaserJet 4 and 4M Printers User's Manual, HP Part No. C2001-90912).

4.4.2 Procedures for 4 or 8mm SCSI Tape Drive Maintenance

On a weekly basis, clean the tape drive heads on the AMHS Server. (Refer to the Sun Tutorial on 4 or 8mm Small Computer Systems Interface (SCSI) Tape Drives for tape head cleaning procedures.

4.5 SYSTEM MONITORING

This section deals with procedures that allow the System Operator to receive information on the status of the processes, backside queues, message feeds and disks.

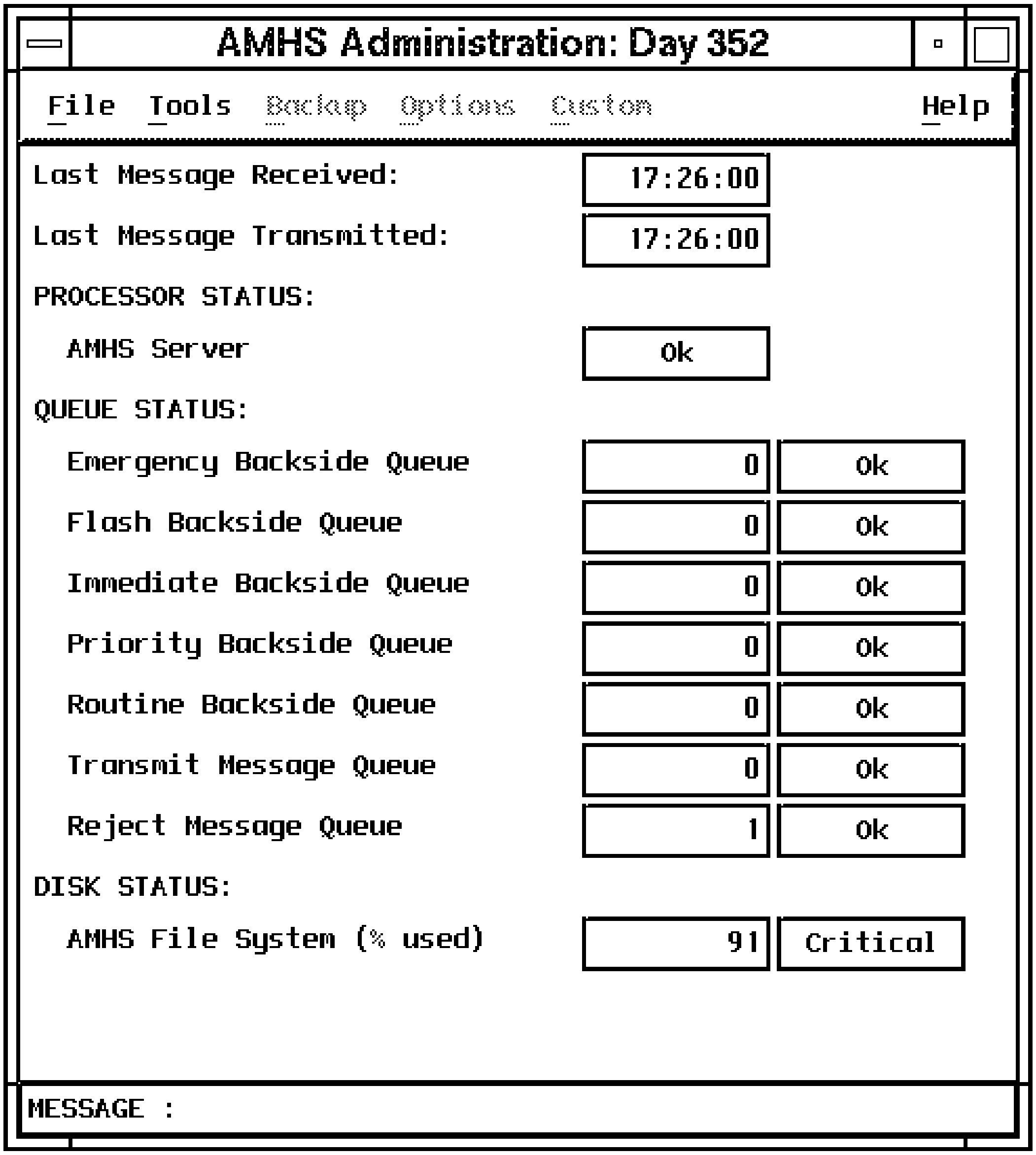
4.5.1 AMHS Administration Status Monitoring Tool

The status monitoring tool is typically displayed on the AMHS Server Console but can be displayed from any workstation or console. The display tool should be configured by the AMHS System Administrator to reflect the monitoring needs of the site. The tool is launched from the Launch System Administration Tools icon and operates with the privileges of the **amhs_dba**, to ensure access to key AMHS areas. The menu bar presents access to additional tools, and operators should take care not to use these additional tools unless the need arises and the operator is trained in their use.

To ensure accurate audit trails, only one user account should have the **amhs_dba** privilege at a time. The AMHS System Administrator will ask the System Operator to close the monitoring tool window when administration tasks are being performed.

The status monitoring tool (Figure 4-1) displays four groups of status information:

(1) Last message received/transmitted is a real-time display to the message time stamp and is useful to see if messages are flowing to and from the SAT.



```
Last Message Received:            17:26:00
Last Message Transmitted:         17:26:00
PROCESSOR STATUS:
  AMHS Server                     Ok
QUEUE STATUS:
  Emergency Backside Queue               0   Ok
  Flash Backside Queue                   0   Ok
  Immediate Backside Queue               0   Ok
  Priority Backside Queue                0   Ok
  Routine Backside Queue                 0   Ok
  Transmit Message Queue                 0   Ok
  Reject Message Queue                   1   Ok
DISK STATUS:
  AMHS File System (% used)             91   Critical

MESSAGE :
```

Figure 4-1. Status Monitoring Tool

(2) Processor Status tells whether the AMHS Server is functioning properly and whether its processes are running in a timely manner. The processor status indicator displays:

(a) Green: All processors and processes are up and running properly.

(b) Yellow: All processors are up and running but one or more processes are down.

(c) Red: The processor is down.

(d) Orange: Bad Configuration. Errors in the Main Window ini file or improper tool installation.

(3) Queue Status displays how many messages are backed-up in each of the queues awaiting processing. If the threshold for the queues is exceeded on any of these queues, the feed has probably stopped and the status indicator will turn to red. Restarting the feed will normally clear this alarm.

(4) Disk Status displays the percent of the total disk space utilized on each monitored drive. The threshold will give early warning of a potential disk full error. Archiving the oldest messages by Julian day will free up disk space.

4.5.2 AMHS Processes Status

Perform the following to determine status of the critical AMHS processes:

(1) From an Xterm, log into the AMHS Server as the **amhs_dba** by typing:

```
rlogin  amhserver  -l  amhs_dba
```

(2) To bring up the database administration menu, type:

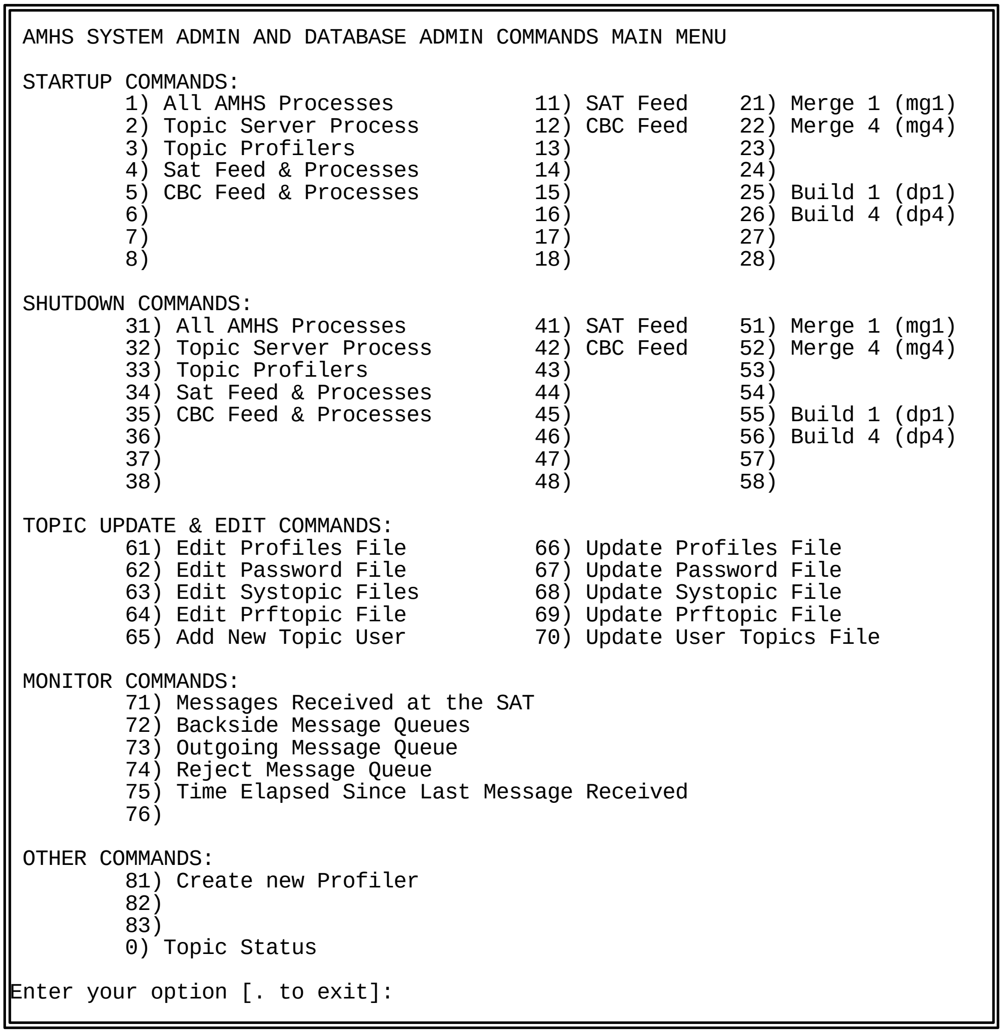
```
topic_cmd
```

(3) Select option “**0**” (**Topic Status**). A listing of the Topic processes is scrolled on the display. Each process listed is considered to be up unless the status line specifically states it is **down**.

(4) View the Summary screen on the Sys Admin Tools main window.

(5) Use the EM System Monitor tool (reference Appendix C, Section C.3.9).

4.5.3 Backside Queue Checks



```
AMHS SYSTEM ADMIN AND DATABASE ADMIN COMMANDS MAIN MENU

STARTUP COMMANDS:
        1) All AMHS Processes           11) SAT Feed    21) Merge 1 (mg1)
        2) Topic Server Process         12) CBC Feed    22) Merge 4 (mg4)
        3) Topic Profilers              13)             23)
        4) Sat Feed & Processes         14)             24)
        5) CBC Feed & Processes         15)             25) Build 1 (dp1)
        6)                              16)             26) Build 4 (dp4)
        7)                              17)             27)
        8)                              18)             28)

SHUTDOWN COMMANDS:
        31) All AMHS Processes          41) SAT Feed    51) Merge 1 (mg1)
        32) Topic Server Process        42) CBC Feed    52) Merge 4 (mg4)
        33) Topic Profilers             43)             53)
        34) Sat Feed & Processes        44)             54)
        35) CBC Feed & Processes        45)             55) Build 1 (dp1)
        36)                             46)             56) Build 4 (dp4)
        37)                             47)             57)
        38)                             48)             58)

TOPIC UPDATE & EDIT COMMANDS:
        61) Edit Profiles File          66) Update Profiles File
        62) Edit Password File          67) Update Password File
        63) Edit Systopic Files         68) Update Systopic File
        64) Edit Prftopic File          69) Update Prftopic File
        65) Add New Topic User          70) Update User Topics File

MONITOR COMMANDS:
        71) Messages Received at the SAT
        72) Backside Message Queues
        73) Outgoing Message Queue
        74) Reject Message Queue
        75) Time Elapsed Since Last Message Received
        76)

OTHER COMMANDS:
        81) Create new Profiler
        82)
        83)
        0) Topic Status

Enter your option [. to exit]:
```

Figure 4-2. topic_cmd Xterm Windows

Backside queues (BSQs) are normally displayed on the Sys Admin Tools main window.

The following information is included for reference only. A BSQ is a queue used by the SAT to store a token for, and describing, each incoming message. There are five (5) BSQs, one for each level of message precedence; **bsq1** for EMERGENCY (Y) messages, **bsq2** for FLASH (Z) messages, **bsq3** for IMMEDIATE (O) messages, **bsq4** for PRIORITY (P) messages, and **bsq5** for ROUTINE (R) messages. To list the contents of all BSQs, execute the following:

(1) **cd /h/AMHS/Server/sat/autodin**

(2) **ls bsq?**
The system will respond by displaying the following:

```
bsq1: x1
bsq2: x2
bsq3: x3
bsq4: x4
bsq5: x5
```

where x1 through x5 represent the number of messages in the respective queue.

If more than 10 entries appear in any of the five BSQs, there may be a system problem. The following explains how to examine the feed activity logs.

4.5.4 Message Feed Checks

Each message source has its own incoming message feed process. The message feed process names are as follows:

sat_feed:	Processes all incoming AUTODIN traffic.
cbc_feed:	Processes all released comeback copy (CBC) traffic.

The primary method for checking a message feed process is through the message feed activity log or “log” file. Each feed process maintains a detailed daily activity log of events. These logs are stored in the **/h/AMHS/Server/usr/topic/amhs_db/log** directory and have a filename that is composed of the name of the feed followed by the log’s Julian day. For example, the **sat_feed** log for Julian 320 would be named **sat_320.log**. Similarly, the **cbc_feed** log for day 123 would be named **cbc_123.log**.

Each feed operates by polling and processing an incoming BSQ. In other words, the feed process monitors the queue waiting for the external interface (message source) to deposit a token. Once the feed process detects the token, the feed process performs some specific tasks based on the message source. After these tasks are performed, the message is passed to the topic merge process (mg). As each of these tasks is performed by the feed process, a log entry is placed in the corresponding “log” file.

The best way to monitor a suspect feed process is by monitoring the log. For example, the following commands will display the current status of the comeback feed:

(1) Determine the current Julian day by typing:

```
date +%j
```

(Assume it is day 043 for purposes of this explanation.)

(2) Show log in real-time by typing:

```
tail -f /h/AMHS/Server/topic/amhs_db/log/cbc_043.log
```

Once these commands are executed, the comeback feed log for day 043 will begin displaying any new log information. If the feed is working properly, new information will be displayed as **cbc_feed** processes messages, in this case, whenever a new message is released and archived. The same information and commands apply for the other feeds.

4.5.5 Topic Database Checks

Each topic database process has its own set of log files. These log files are maintained in the **/h/AMHS/Server/topic/amhs_db/log** directory. Each log file is named with the process named followed by an "adt" extension. For example, the first topic merger process log is named **mg1.adt** .

As a topic database process handles incoming messages, entries are placed in the corresponding log file. Each TOPIC database process is triggered by one of the message feed processes (i.e., the correct operation of a TOPIC database process is dependent on its message feed process). The mapping of the message feed process to the topic database process is as follows:

Table 4-1. TOPIC Database Processes

Source	Feed Process	Data Prep	Merge Process
AUTODIN	sat_feed	dp1	mg1
Comeback	cbc_feed	dp4	mg4

To display the merger log for AUTODIN messages in real-time, list the log by executing the following command:

```
tail -f /h/AMHS/Server/topic/amhs_db/log/mg1.adt
```

4.5.6 Disk Status

The following are UNIX disk status commands for the Sun workstations.

df -k This command is issued from an Xterm window. It shows all the disks available on a machine and what directories (file systems) they are mounted on. It also shows usage in kilobytes (KB) and how many KBs are still available in each file system (disk usage). This is the same as the Disk Status feature in Figure 4-1.

fsck This command checks and repairs file systems. This command lists each file system on the disk as it is checked and states if there are any problems. Some problems can self-correct. If this happens, the system will state so. If not, it will report the errors. Only use this command if the SOPs authorize its use.

NOTE: If file systems have trouble during manual mounting or at bootup, this command could fix the problem.

4.6 BACKUPS

An effective backup strategy is an essential component of the disaster recovery plan. The backup plan developed as part of the site SOPs should consider the time and effort to recover from a system failure and how important it is to recover all or most of the data accumulated in the system. One possible scenario is to do full backups monthly and incremental backups daily. The restore process involves reloading each tape in sequence starting from the last full backup and then manually rebuilding the data accumulated since the last incremental backup. This requires repeating all transactions that occurred during the time between the last backup and the system failure. In the AMHS environment this can have a significant effect on system down time and require the users to reprocess all of the incoming messages that are refed from the AUTODIN switching center archives, not to mention constructing all the message traffic originated during the time interval to completely restore the message database.

Full backups (Level 0) are difficult because they require that all the AMHS users close all AMHS activities, all TOPIC processes be stopped, and that the server be put in single user mode to ensure that no data or transactions are lost during backup. If any users have not closed their AMHS windows, the server shutdown will close them and unsaved data will be lost. Incremental backups are less strict, but a time must be selected when user activity is very low and the SAT can easily be put off-line temporarily. It is also important that no crontabs are running and that no System Administration activities are being performed.

The recommended AMHS backup strategy is to perform a full Level 0 backup of the system **weekly**, preferably just before start of the first day shift on Monday morning or during the least active time for your site. The users should be instructed to close all AMHS activities at this time or at the end of their previous shift if the workstation is not occupied 24 hours a day. Incremental backups should be performed **daily** (preferably Level 5). The EM audit logs and Monitor and Control status will indicate this has happened.

4.6.1 Sun Server Backup Procedure

A Level 0 backup is performed when the complete system is to be backed up. This kind of backup saves everything regardless of previous backups. During a Level 0 backup, the operator puts the system into single user mode, during which, the server being backed up is not available to users and any open AMHS user windows will be closed. The steps to perform a Level 0 backup are:

(1) Type: **fsck**

This is not an essential step, but it checks the disk to be backed up and corrects problems that can cause the backup to fail. (Recommendation: Monitor output for failures.)

(2) Type: **df -k | lp**

Sends a copy of the filesystem names to the printer. This copy should be saved with the tapes so these filesystems can be recreated after a crash.

(3) Type: **m -f /"device name" status**

The “device name” is the name of the tape drive you intend to use and check if the tape drive is working. A blank tape must be installed in the drive to do this test.

Bring the system down to single user mode (if no users are currently on the server). To bring the system down in two minutes:

(4) Type: **sync**
Type: **shutdown -y -g120 -i0**

At the **OK>** prompt, boot to single user mode, e.g. at **OK>** type:

boot -rs

(5) To perform a Level 0 backup, for each partition on the filesystem printout except /swap, /tmp, and filesystems mounted from other machines. Insert a new blank tape in the tape drive and run the following.

Type: **/usr/sbin/ufsdump 0ucf /”tape device name" /"disk partition name"**

(6) Label each tape with information consistent with your site.

Host name =sun1	
Dump level = 0	
Volume	**1 of 3**
Partition	**c0t3d0s0**
Date	Dec 25, 1995

(7) From the single user mode, the following will boot the machine back to full user mode:

Type: **sync**
Type: **shutdown -y -g0 -i0**

At the **OK>** prompt, boot to full user mode, e.g. at **OK>** :

Type: **boot**

(8) To perform a level [1-9] incremental backup on single tape, insert a new blank tape in the tape drive and run the following for each partition.

Type: **/usr/sbin/ufsdump [1-9]ucf /"tape device name" /"disk partition name"**

4.6.2 Server Backup Procedure Strategy

Figure 4-3 demonstrates the idea behind the Level 0/Level 5 backup. Showing the recommended backup strategy. Backup Levels 1-9 are incremental, that is, they back up only those files that changed since the next lower backup Level. If a failure occurs in the location illustrated in the above figure, the strategy would be to restore the last Level 0 backup and then restore the most recent Level 5 incremental backup. (For additional information see the Sun manual for Solaris 2.3, Administrating File Systems, Chapters 8, 9, and 10.) It is important to use the Sun Solaris standard backup processes and it is equally important to consider the restoration time and the possibilities of losing messages when you develop your site specific backup SOP's

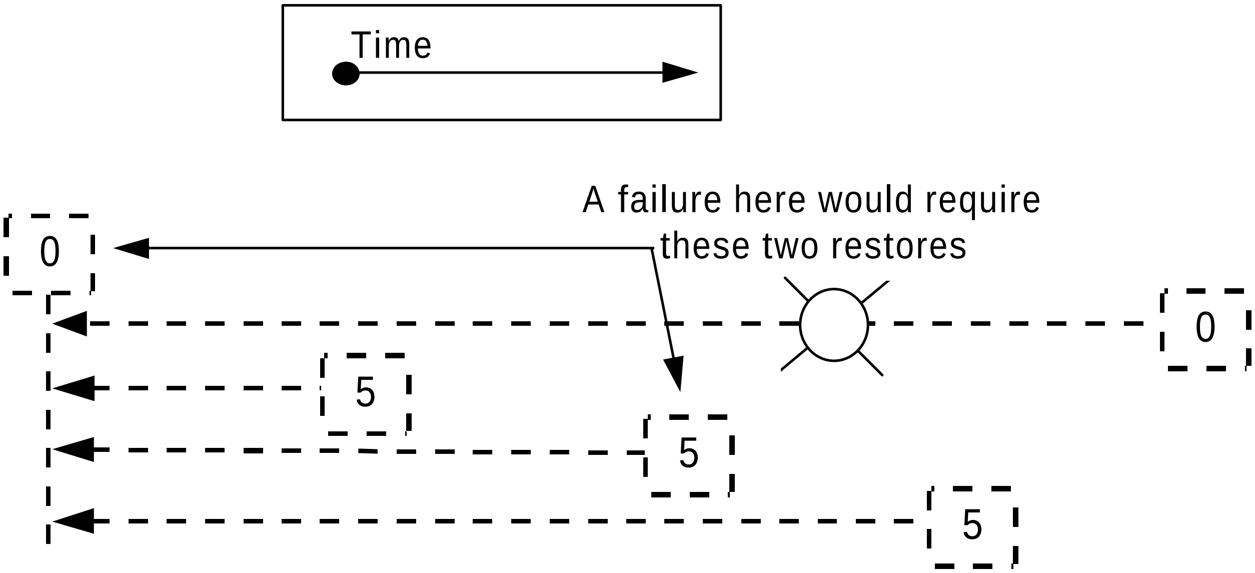


Figure 4-3. Sun Server Backup Strategies

4.6.3 Workstation Backup Strategy

The user workstations do not contain any AMHS real-time data. This data is all stored via NFS mounts to the servers. After a user's workstation is configured and operational, a Level 0 backup of the workstation will contain all the information to restore that user workstation. Other client workstation applications may require different backup strategies. These strategies will not affect the AMHS functionality of the workstation, if all of the directories are restored and the softlinks and NFS mounts are restored.

This page intentionally left blank.